Adaptive Geospatial Data Representation for Data Analytics and Sharing in the Mobility Domain

Kumulative Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von
Rajjat Dadwal
aus
Mangarh, Kangra, India

Bonn, 2024

Angefertigt mit Genehmigung
der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachterin/Betreuerin: Prof. Dr. Elena Demidova
Gutachter: PD Dr. Volker Steinhage

Tag der Promotion: 17.01.2025
Erscheinungsjahr: 2025

ABSTRACT

In recent years, the mobility domain has gained attention from urban planners and researchers due to its essential role in enhancing urban safety and development. This interest can be attributed to the increased availability of geospatial and mobility data from a wide variety of sources, such as OpenStreetMap and knowledge graphs. Geospatial and mobility data enable the development of predictive models such as accident and crime prediction, enhancing urban safety, and planning. However, there are specific challenges to utilizing geospatial and mobility data when building predictive models. First, mobility data is typically sparse. Data sparsity occurs when spatio-temporal events, such as traffic accidents, are scarce and scattered across geographic regions. Due to data sparsity, predicting future events at specific locations becomes challenging. Second, geospatial and mobility data from multiple sources are often utilized by machine learning pipelines to generate latent representations. The latent representations derived from multimodal data are richer in context and beneficial for several predictive tasks. However, the diversity in data sources makes it challenging for machine learning pipelines to integrate these sources effectively, resulting in ineffective latent representations. Third, personal mobility data can contain sensitive information, such as visited locations, traveled routes, and driver profiles. Applications relying on personal mobility data require effective and robust methods to confirm provenance and authenticity. However, existing methods in the mobility domain are neither effective nor robust, which makes tracing personal mobility data challenging. This lack of traceability of personal mobility data limits its use in predictive model development.

This cumulative thesis summarizes several novel methods to address these challenges. First, we propose a novel adaptive clustering method for accident prediction (*ACAP*) to address the challenge of data sparsity. *ACAP* aggregates traffic accident events dynamically with a grid-growing algorithm while considering underlying data distribution. Furthermore, *ACAP* enhances the prediction results of traffic accident events by focusing on adaptive task-specific regions. Second, to address the challenge of ineffective latent representations of geospatial regions, we propose a multimodal and multitask approach for region representation learning (*MAGRE*). *MAGRE* leverages multitask learning combined with attention-based fusion to enhance the effectiveness of region latent representations. These effective latent representations maintain the semantics for several downstream predictive tasks. Furthermore, the adaptive representations generated by *MAGRE* can be aggregated for user regions of interest of any shape and size without retraining. Third, to address the challenge of the lack of traceability of personal mobility data, we propose a novel watermarking approach for GPS trajectories called *W-Trace*. *W-Trace* embeds watermarks within GPS trajectories and is robust to adversarial modifications, enhancing traceability. In addition, *W-Trace* maintains the utility of watermarked GPS trajectories for several downstream tasks. In summary, this thesis presents three novel contributions: i) an adaptive aggregation method for accident event data, ii) an effective and adaptive representation learning approach for geospatial regions, and iii) an effective, robust, and utility-preserving watermarking method for GPS trajectories.

Keywords: *Spatio-temporal Data Analysis, Adaptive Clustering, Adaptive Geospatial Region Representation, Watermarking GPS Trajectory*

Acknowledgements

First and foremost, I am deeply thankful to my supervisor, Prof. Dr. Elena Demidova, for her invaluable guidance and support during my Ph.D. I also wish to express my warm and sincere gratitude to my second reviewer, PD Dr. Volker Steinhage, for his insightful comments on the draft of my thesis. I thank Prof. Dr. Thomas Schultz and Prof. Dr. Jochen Dingfelder for being a part of the doctoral committee.

A special thanks to Dr. Thorben Funke and Dr. Nicolas Tempelmeier for their guidance during the early stages of my Ph.D. I am grateful to Dr. Ran Yu for her guidance during my PhD journey and constructive feedback on my thesis draft. I also thank Dr. Michael Nüsken for the productive discussions during the paper submissions. I would also like to acknowledge the support and encouragement of my former colleagues from the L3S Research Center, Hannover, including Ashutosh, Stefan, Sara, and Tin. My thanks also go to my DSIS group colleagues, Alishiba, Marco, Uttam, Steve, and Genivika, for their support along the way. Finally, I am deeply grateful to my family, especially my parents, for their unwavering support. My heartfelt thanks go to my wife, Kokila, for standing by me throughout this journey.

The works presented in this thesis were partially funded by the Federal Ministry for Economic Affairs and Climate Action (BMWK), Germany, (("CampaNeo", 01MD19007B), ("d-E-mand", 01ME19009B), ("ATTENTION!", 01MJ22012C)), the European Commission (EU H2020, "smashHit", 871477), the German Research Foundation ("WorldKG", 424985896), and the Federal Ministry for Digital and Transport (BMDV) ("MoToRes", 19F2271C) and the Lamarr Institute for Machine Learning and Artificial Intelligence.

List of Publications

The Chapters 4-6 are based on the following research publications.

In Chapter 4, we propose a novel adaptive clustering method for accident prediction to tackle the data sparsity challenge. This method creates adaptive geospatial clusters that follow the underlying data distribution and perform traffic accident predictions in these clusters.

[DFD21] **Rajjat Dadwal**, Thorben Funke, and Elena Demidova. "An Adaptive Clustering Approach for Accident Prediction". In Proceedings of the 24th IEEE International Intelligent Transportation Systems Conference, ITSC 2021. IEEE, 2021, pp. 1405-1411. DOI: 10.1109/ITSC48978.2021.9564564

In Chapter 5, we propose an effective and adaptive latent representation learning method for geospatial regions. The latent representations generated by the proposed method effectively capture the semantics from different modalities and can be aggregated to any region of interest.

[DYD24] **Rajjat Dadwal**, Ran Yu, and Elena Demidova. "A Multimodal and Multitask Approach for Adaptive Geospatial Region Embeddings". In Proceedings of the 28th Pacific-Asia Conference on Knowledge Discovery and Data Mining Conference, PAKDD 2024. Springer, 2024, pp. 363-375. DOI: 10.1007/978-981-97-2262-4_29

Then, in Chapter 6, we propose an effective, robust, and utility-preserving watermarking approach for GPS trajectories to address the lack of *traceability* challenge in personal mobility data.

- [Dad+24] **Rajjat Dadwal**, Thorben Funke, Michael Nüsken, and Elena Demidova. "Towards effective, robust and utility-preserving watermarking of GPS trajectories". Accepted for publication in ACM Transactions on Spatial Algorithms and Systems, TSAS, accepted on 03 October 2024. DOI: 10.1145/3701558
- [Dad+22] **Rajjat Dadwal**, Thorben Funke, Michael Nüsken, and Elena Demidova. "W-trace: robust and effective watermarking for GPS trajectories." In Proceedings of the 30th International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2022. ACM, 2022, pp. 77:1–77:4. DOI: 10.1145/3557915.3561474 (Short paper)

Furthermore, I have contributed to a publication that is not part of the thesis.

[Wow+22] Kelvin Sopnan Wowo, **Rajjat Dadwal**, Timo Graen, Andrea Fiege, Michael Nolting, Wolfgang Nejdl, Elena Demidova, and Thorben Funke. "Using

Vehicle Data to Enhance Prediction of Accident-Prone Areas.” In Proceedings of the 25th IEEE International Conference on Intelligent Transportation Systems Conference, ITSC 2022. IEEE, 2022, pp. 2450-2456. DOI:10.1109/ITSC55140.2022.9922236

List of Figures

Contents

Chapter 1

Introduction

This chapter begins with introducing geospatial and mobility data and discusses its importance in predictive modeling. Then, we explore several challenges associated with this data for building predictive models. Next, we outline the key research questions to address these challenges. In the end, we summarize the main contributions of the thesis.

1.1 Motivation

Mobility is a basic human trait. It is a fundamental behavior that drives us to move from one place to another to fulfill our needs. Thousands of years ago, humans traveled long distances for food and shelter, motivated by their need for survival. Over time, technological advancements in sensors have pushed the mobility domain to new heights. Nowadays, most vehicles are equipped with sensors that capture various parameters, such as speed, acceleration, and steering wheel angle, generating a wealth of mobility data. Beyond physical movement, mobility also requires an understanding of the geography of a location. In this context, geographic information plays an important role in understanding the surroundings of a location, which is beneficial for navigating unfamiliar places. One such example of geographic information is geospatial maps. Geospatial maps have also transitioned from paper-based to digital maps, such as OpenStreetMap (OSM)[1] [OSMa], producing vast amounts of geospatial data. Geospatial and mobility data acquired from multiple sources enables the development of predictive models in the mobility domain. For instance, the predictive models can help in predicting traffic accidents [DFD21], and support improved urban planning and safer transportation systems.

Utilizing geospatial and mobility data for predictive modeling comes with challenges. The first challenge is related to the sparsity of mobility data. This sparsity occurs when events like traffic accidents are scarce and dispersed across geographical regions. As a result, predicting future traffic accident events at specific locations becomes challenging. The second challenge involves integrating geospatial and mobility data from multiple sources to enhance the learning context for machine learning (ML) pipelines. These machine learning pipelines create latent representations for geospatial regions from multimodal data, which can be beneficial to several predictive tasks such as crime prediction and land use classification. However, the diversity in data sources makes it difficult for ML pipelines to integrate diverse sources effectively, resulting in ineffective region representations. In addition, the broad range of user regions of interest (ROIs) further complicates learning effective

[1] The OpenStreetMap name is a trademark of the OpenStreetMap Foundation and is used with their permission. We are not endorsed by or affiliated with the OpenStreetMap Foundation.

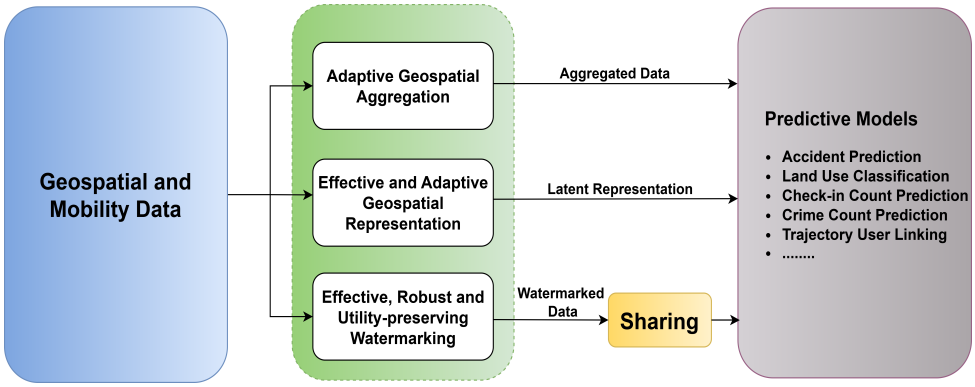


FIGURE 1.1: Overall pipeline for building predictive models utilizing geospatial and mobility data adopted in this thesis

representations for geospatial regions. The third challenge is the sensitive nature of personal mobility data, which can include details such as visited locations or driver profiles. Applications that rely on personal mobility data need effective and robust methods to verify provenance and authenticity. However, existing methods in the mobility domain are neither effective nor robust [Pan+19; Jin+05], which makes tracing of personal mobility data challenging. This lack of traceability limits the use of personal mobility data in predictive model development.

This cumulative thesis addresses the challenges associated with geospatial and mobility data in building predictive models. In particular, we address three main challenges: i) data sparsity, ii) ineffective latent representations, and iii) lack of traceability. Figure 1.1 illustrates an overall pipeline of building predictive models utilizing geospatial and mobility data. In this figure, we illustrate how the above-mentioned challenges are addressed in this thesis by three novel methods, which generate relevant data representations, such as aggregated data, effective latent representations, and watermarked data. In the end, these representations are utilized in several prediction tasks.

1.2 Research Questions

In this section, we discuss the challenges associated with geospatial and mobility data, i.e., data sparsity, ineffective latent representations, and lack of traceability, which are addressed in this thesis.

RQ1. How to create an adaptive geospatial aggregation method to predict traffic accidents in urban regions?

A traditional approach to address data sparsity in the mobility domain relies on fixed geospatial aggregations, such as fixed grids or administrative boundaries [Moo+19; Zha+20]. These fixed aggregations aggregate the data from multi-modalities within a grid or administrative boundary and perform predictions on these spatial aggregations. However, these fixed aggregations often fail to align with the actual spatial distribution of the event data [DFD21]. For instance, traffic accident events in a given region might be split across multiple grid cells, leading to a few events in each grid cell. Furthermore, there is no standard approach for selecting the fixed aggregation grid size, such as 1km $\times$ 1km or 5km $\times$ 5km. This inconsistency can result in variations in prediction results in traffic accident prediction tasks. Therefore, addressing the data sparsity challenge requires adaptive

aggregation methods for event data that accurately reflect the underlying spatial distribution of the data. Geospatial and mobility data from multi-modalities are often utilized by machine learning pipelines to create geospatial region representations. Due to data heterogeneity, creating effective and adaptive latent representations for geospatial regions is challenging. This brings us to the second research question.

RQ2. How to create an effective and adaptive geospatial region latent representation?

Geospatial and mobility data from multiple sources are often utilized to enhance contextual understanding of machine learning pipelines. These ML pipelines create region representations from multimodal data that are beneficial to several predictive tasks. Due to data heterogeneity, integrating multiple data sources with different structures and types is challenging for ML pipelines, leading to ineffective region representations. Furthermore, various user regions of interest require adaptive region representations. For instance, one might be interested in predicting the crime rate within a 200-meter radius of Bonn Central Station, while another might want to extend this radius to 500 meters. A machine learning pipeline trained for one region of interest requires retraining for another, increasing the computational costs of training. However, creating effective and adaptive geospatial latent representations that do not require retraining for ROIs and can be utilized for several downstream predictive tasks is challenging. Therefore, there is a need to develop an effective and adaptive geospatial latent representation learning method that can retain the semantics from multimodal data for several predictive tasks and flexibly align with user-defined ROIs.

Creating latent representations for geospatial regions from mobility data, such as GPS trajectories, can include user-specific information. Once this data is shared, authenticating the shared data and verifying its provenance becomes challenging. Verifying the data provenance is important in several scenarios, such as verifying user consent, confirming the driver's identity during risk assessment for personalized insurance policies, and validating insurance claims [Dad+24]. This brings us to the third research question.

RQ3. How to develop a robust, effective, and utility-preserving watermarking method for GPS trajectories?

As discussed, GPS trajectory data may contain personal information such as travel patterns and driver profiles. Sharing such data for any task needs careful handling [Dad+22]. To address the challenge of lack of traceability, provenance information can be embedded in GPS trajectories with a watermarking method, resulting in watermarked data. Watermarking integrates provenance information within the trajectory data, enabling the tracing of data origin. However, developing an effective, robust, and utility-preserving watermarking technique for GPS trajectories poses several challenges. On the one hand, a watermark needs to be effective and robust. This means the watermark should embed enough information for verification and be resilient against modifications by potential adversaries. On the other hand, the watermark should have minimal impact on the utility of watermarked trajectories for downstream applications. Additionally, GPS trajectories present unique challenges for digital watermarking due to positional inaccuracies and non-uniform sampling rates. Therefore, there is a need to develop a novel watermarking approach for GPS trajectories to enable the watermark to remain intact under various data modifications while maintaining the data utility for analytical and predictive tasks.

1.3 Contributions

This section presents our contributions to address the research questions presented above.

Adaptive Clustering for Aggregating Accidents Events [DFD21]. As discussed in RQ1, to tackle the data sparsity challenge, spatio-temporal data, such as traffic accident events, are often aggregated into fixed geospatial aggregations, which leads to uneven data distributions. To address these challenges, we propose a novel adaptive clustering method for accident prediction (*ACAP*), which identifies task-specific regions for prediction purposes. The adaptive clusters generated with the grid-growing algorithm capture the underlying spatial distribution of the traffic accident events, addressing the problem of uneven distributions. We utilize a neural network architecture to predict accident events within adaptive clusters at specific time frames. The experiment results demonstrate that the *ACAP* approach increases the F1-score by 2-3 percent point on average compared to existing state-of-the-art methods in three German cities. Furthermore, the grid-growing approach outperforms the clustering-based baselines by four percentage points on average. The feature analysis experiment indicates the importance of points of interest (POIs) and temporal features in improving traffic accident prediction results.

Adaptive Geospatial Region Latent Representation [DYD24]. To address the challenge of ineffective latent representations in geospatial data representation learning (as outlined in RQ2), we introduce *MAGRE*: a novel approach designed to create effective and adaptive region representations. By incorporating features from multi-modal sources, such as OSM images and POI count, *MAGRE* enriches the context for the region representation learning process. Furthermore, multitask learning based on an attention-based fusion effectively integrates different data representations and enhances the effectiveness of region representations. The effectiveness of the *MAGRE* representations is evaluated on several downstream tasks. In particular, the experimental results demonstrate that the *MAGRE* approach outperforms state-of-the-art embedding baselines, reducing root mean squared error by 19.08% for check-in count prediction and by 25.73% for crime rate prediction. The use case study on crime prediction task demonstrates that the region embeddings generated by *MAGRE* can handle ROIs of different shapes and sizes, demonstrating the adaptiveness of our approach.

Robust, Effective, and Utility-preserving Watermarking Method [Dad+24; Dad+22]. To address the lack of traceability challenge (as outlined in RQ3), we present *W-Trace*, a novel GPS watermarking method that is robust, effective, and preserves utility for downstream tasks. *W-Trace* transforms the GPS trajectories into a complex domain, and Fourier transformation is applied to decompose the trajectory into frequency representation. The watermark is embedded in these frequency components and verified through a spatiotemporally-aware procedure. The GPS trajectories watermarked by the *W-Trace* method are robust to several modifications, achieving a high recognition rate of 99% on average on two datasets. This high recognition rate highlights the robustness and effectiveness of the *W-Trace* method, enabling data traceability. *W-Trace* preserves the utility of watermarked GPS trajectories for downstream tasks like map matching and predictive tasks such as trajectory user linking. Furthermore, *W-Trace* embeds more watermark information into the GPS trajectories than the state-of-the-art methods.

1.4 Thesis Structure

The rest of the thesis is structured as follows:

- Chapter 2 provides an overview of relevant concepts crucial for understanding the terminology adopted in the thesis. This chapter introduces a geospatial and mobility data-based predictive pipeline that illustrates various concepts, such as spatio-temporal data representations, aggregations, and predictive model development.
- In Chapter 3, we discuss the state-of-the-art works for event prediction, latent representations for geospatial regions, and watermarking methods.
- Chapter 4 addresses the first research question on data sparsity and summarizes the proposed adaptive clustering approach for traffic accident prediction.
- Chapter 5 addresses the second research question regarding ineffective latent representations and introduces a novel method for generating effective and adaptive latent representations for geospatial regions.
- Then, Chapter 6 addresses the third challenge regarding the lack of traceability and discusses an effective, robust, and utility-preserving watermarking approach for GPS trajectories.
- Lastly, Chapter 7 concludes our work by briefly summarizing the results obtained throughout the thesis. Additionally, this chapter provides an outline for potential future research directions.

Chapter 2

Background

This chapter presents fundamental concepts for understanding the terminology adopted in the thesis. We introduce a predictive pipeline based on geospatial and mobility data, beginning with data acquisition from different sources, followed by a series of transformations, and concluding with developing predictive models.

Geospatial and mobility data come in different types and granularity over space and time. As discussed in Chapter 1, this data is beneficial in several predictive tasks, including traffic prediction, accident prediction, and travel time estimation [Moo+19; DFD21]. Geospatial and mobility data often require various transformations to make the data suitable for modeling. Figure 2.1 illustrates typical components of a geospatial and mobility data-based predictive pipeline. This pipeline begins by acquiring geospatial and mobility data from different sources. Next, the data is transformed into spatio-temporal (ST) representations, such as geospatial points and trajectories. To enable traceability, the mobility data, such as GPS trajectories, is processed through a watermarking method where provenance information is embedded into the data. Then, the data aggregation step transforms the data to create different aggregations, addressing the data sparsity challenge. Next, the model development step transforms the aggregated data into meaningful representations for learning models. This step then identifies the most suitable predictive model for the generated representations. Finally, the selected model outputs predictions based on the tasks, such as regression and classification. We discuss each step in more detail in the following sections.

2.1 Geospatial and Mobility Data

This section explores data sources responsible for generating geospatial and mobility data, as illustrated in the first block of Figure 2.1.

2.1.1 Sources of Geospatial Data

Geographic information, such as the location details of geographic entities like Points of Interest (POIs), is referred to as *geospatial data*. Satellite imagery, volunteered geographic information (VGI), web-based applications, and knowledge graphs (KGs) are some sources that contain geospatial data. In this thesis, we utilize geospatial data from VGI. VGI refers to geographic data collected through the efforts of volunteers. A well-known example of VGI is OpenStreetMap (OSM). OSM is an open-source spatial database that aims to capture data about geographic objects like roads, rivers, and country boundaries. OSM operates under the Open Database License (ODbL), with contributors voluntarily providing geographic data.

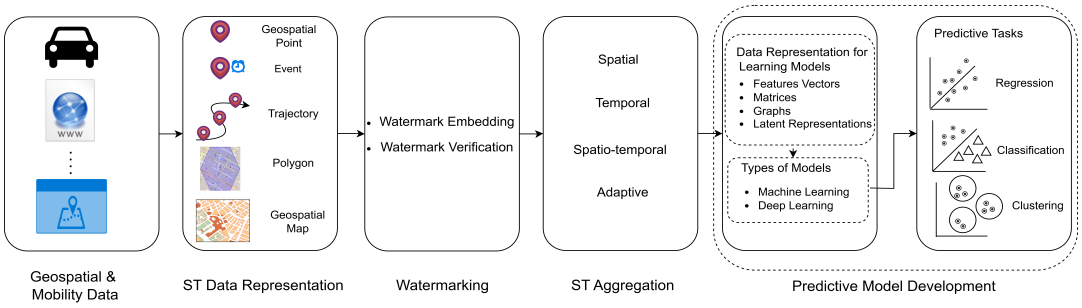


FIGURE 2.1: Overview of a geospatial and spatio-temporal (ST) mobility data-based predictive pipeline. Map data: ©OpenStreetMap contributors, ODbL

OSM models spatial data in three ways: *nodes*, *ways*, and *relations*. A *node* is a specific geographic point on the earth's surface, e.g., landmarks. Each *node* is characterized by a unique identifier and its corresponding latitude and longitude coordinates. *Ways* are represented by a sequence of nodes, illustrating entities such as roads or rivers. *Relations* represent a complex object that is an ordered list of nodes, ways, and other relations, such as city boundaries or administrative districts. The different ways to access geospatial data from OSM are:

- **Nominatim** leverages the OSM data to perform geocoding, enabling users to search for locations based on names and addresses [Nom].
- **OSM API** is an interface for accessing and modifying geospatial data stored within the OSM database [APIa].
- **Overpass API** is a read-only API to access specific segments of the OSM data and works similarly to a web-based database [APIb].
- **OSM Planet** consolidates the OSM data into a single file released weekly, with each new version [OSMb].

In this thesis, we access geospatial data from OSM, which is helpful in predictive tasks, such as land use classification and crime prediction [DYD24].

2.1.2 Sources of Mobility Data

Mobility data is often represented with space and time. Mobility data originates from GPS sensors, transmitting data periodically over time [An+16], as illustrated in the first block of Figure 2.1. For instance, a GPS sensor installed in a car continuously records the location of the vehicle at specific intervals. Similarly, sharing locations on location-based social networks (LBSNs) at different time intervals leads to the generation of mobility data [Yan+13]. The mobility data has applications in predictive tasks such as next location prediction [Sun+24], crowd flow prediction [Jia+23], region representation [DYD24] and traffic prediction [Han+23].

The geospatial and mobility data acquired from the above sources typically require further preprocessing for predictive modeling. One of the preprocessing steps is transforming the geospatial and mobility data into spatio-temporal data representations.

2.2 Spatio-temporal Data Representation

This step transforms geospatial and mobility data into one of the well-known spatio-temporal data representations, as illustrated in the second component of Figure 2.1. This thesis discusses the most common spatio-temporal data representations, such as geospatial point, event, trajectory, polygon, and spatial map.

2.2.1 Geospatial Point

A geospatial point represents the geographic location of a particular place, represented by a pair of latitude and longitude [DGL07].

Definition 1 (Geospatial Point) *A geospatial point P is a point located on the earth's surface and denoted as:*

$$P = (lat, lon),$$

where lat is the latitude, and lon is the longitude.

For instance, Bonn Central Station is a Point of Interest (POI), represented by a geospatial point $(50.73185, 7.09776)$, where 50.73185 and 7.09776 are latitude and longitude, respectively, as illustrated in Figure 2.2a. The distance between the two geospatial points can be calculated utilizing Euclidean or Haversine distance. Geospatial points are called *nodes* in OSM data modeling, as described in Section 2.1.1.

2.2.2 Event

An event is a real-world occurrence that takes place at a particular time and location [Zha22].

Definition 2 (Event) *An event E is recorded at a specific geospatial point P and time t, denoted as:*

$$E = (P, t).$$

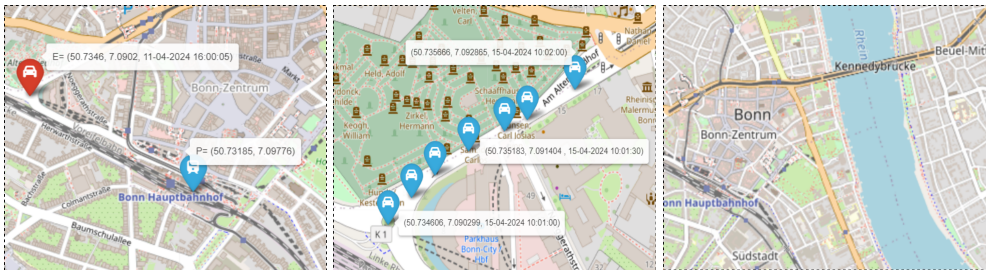
For instance, an accident event is recorded with a geospatial point $(50.7346, 7.0902)$ at time 11-04-2024 16:00:05, as illustrated in Figure 2.2a. In this thesis, we utilize traffic accident data as events and POI data from OSM as geospatial points for the accident prediction task [DFD21].

2.2.3 Trajectory

A trajectory is an ordered collection of geospatial points, where each point is recorded at specific timestamps. The geospatial points in the trajectory represent the locations visited by the moving object, as illustrated in Figure 2.2b. Examples of trajectories include the path followed by a taxi from the point of pick-up to the drop-off destination [Far+16].

Definition 3 (Trajectory) *A trajectory T consists of geospatial points organized chronologically and paired with the corresponding timestamps [Dad+22],*

$$T = [(P_j, t_j)], \textit{with } t_j < t_{j+1} \textit{ for all } j,$$



A) Geospatial point and event B) Trajectory C) Geospatial map

FIGURE 2.2: Examples of spatio-temporal data representations: geospatial point, event, trajectory, and geospatial map. Map data: ©OpenStreetMap contributors, ODbL

where $P_j = (lat_j, lon_j)$ *denotes the geospatial point, and* t_j *refers to the timestamp associated with* P_j.

Further preprocessing steps can enhance the trajectory quality and usefulness in application tasks. For instance, initial preprocessing steps can include data cleaning to remove outliers, matching the GPS coordinates to the road segments (map matching), trajectory segmentation to identify distinct movement patterns, and feature extraction to extract relevant features for the analysis. In this thesis, we leverage trajectory data for watermarking GPS coordinates [Dad+22; Dad+24].

2.2.4 Polygon

A polygon encloses a geospatial region with a specific size and shape. Examples of geospatial regions represented with polygon shapes can be squares, rectangles, and hexagons. Polygons represent geospatial entities such as parks, city boundaries, building footprints, or water bodies. In this thesis, a polygon is defined as follows:

Definition 4 (Polygon) *A polygon* G_r *consists of unique geospatial points connected by straight lines [ESR],*

$$G_r = [(P_1, P_2, ..., P_{n-1}, P_n)],$$

such that the last and first geospatial points are identical, i.e., $P_1 = P_n$.

This thesis utilizes polygon shapes like squares and hexagons for partitioning the geospatial regions in the following tasks: traffic accident prediction and geospatial region representation [DFD21; DYD24].

2.2.5 Geospatial Map

A geospatial map is a visual representation of geographic areas, generally displaying spatial features such as boundaries, landmarks, and geographical attributes. Figure 2.2c illustrates a geospatial map of the Bonn region represented as an OSM image. The OSM utilizes distinct colors to represent different objects, facilitating the interpretation of the map. For instance, the light-brown color commonly depicts buildings, blue represents water bodies, and green signifies trees and vegetation. The encoded colors in the OSM provide information about the region characteristics, which can help learn region representations and improve land use type predictions, such as identifying business and commercial areas. In this thesis, we divide the OSM

into various image segments and utilize the segmented images to learn the latent representation of the geospatial regions [DYD24].

2.3 Watermarking

Spatio-temporal data representations, such as trajectory, can contain sensitive information. Sharing such data needs careful handling [Dad+22]. To address these concerns, inserting provenance information into the data before sharing is essential. This allows the shared data to be traced back to its source, enabling traceability. One such method for embedding provenance information is watermarking. Watermarking refers to techniques that embed provenance information (referred to as a watermark) within data [Dad+22]. There are two types of watermarking approaches: one is blind, and another is non-blind. The blind watermarking techniques do not require the original data for watermark extraction, while the non-blind methods require access to the original data. Generally, the non-blind methods are more robust against attacks than the blind [HKB09]. The watermarking process typically consists of two main steps: watermark embedding and watermark verification. We discuss each of the steps in detail.

2.3.1 Watermark Embedding

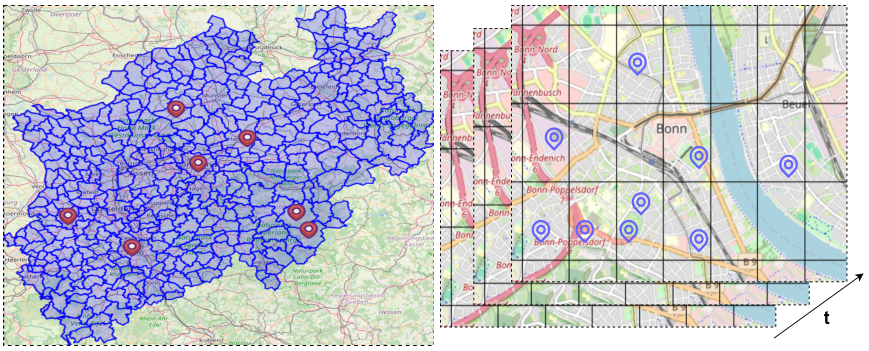
Watermark embedding is the process of inserting a watermark into existing data without affecting the data utility [Dad+22]. The watermark embedding method inserts a watermark into each data sample, which results in the creation of watermarked data. Once the watermark embedding process is complete, the watermarked data can be shared for further analysis. This watermarked data can be modified by adversarial modifications, transforming the data into attacked data. These modifications can pose threats to the authenticity and reliability of the shared data. Watermark verification is crucial to verifying the provenance information in the attacked data.

2.3.2 Watermark Verification

The watermark verification process validates whether the attacked data contains the inserted watermark. The first step in watermark verification is watermark extraction. Watermark extraction is essential in recovering the embedded watermark from the attacked data. Once the watermark is extracted, the extracted watermark needs to be verified against the original inserted one. In this thesis, we utilize Normalized Cross-Correlation (NCC) to find the correlation between two watermarks, which is defined as:

$$\mathrm{NCC}(W, \hat{W}') = \frac{\sum_i W_i \hat{W}'_i}{\sqrt{\sum_i W_i^2}\sqrt{\sum_i \hat{W}'^2_i}},$$

where W is the inserted watermark in the original data and $\hat{W}'$ is the extracted watermark from the attacked data. The value of NCC lies between -1 and 1. The NCC value 1 indicates that watermarks are highly correlated, while 0 and -1 indicate no correlation and negative correlation, respectively. The watermark verification is successful if the NCC score between two watermarks for a given data sample exceeds a predefined threshold. To identify whether a whole dataset is watermarked,



A) Spatial aggregation of traffic accident events based on administrative districts

B) Spatio-temporal aggregation of traffic accident events based on grids

FIGURE 2.3: Examples of spatial and spatio-temporal aggregations of traffic accident events. Map data: ©OpenStreetMap contributors, ODbL

the recognition rate can be utilized as an evaluation metric. The recognition rate is the ratio of correctly identified watermarked samples (true positives) to the total number of watermarked samples. This thesis proposes a robust, effective, and utility-preserving watermarking method for GPS trajectories to address the lack of traceability challenge [Dad+22; Dad+24].

2.4 Data Aggregation

The data aggregation step transforms spatio-temporal representations into an aggregated form to handle the data sparsity. This thesis discusses four main types of aggregations: spatial, temporal, spatio-temporal, and adaptive, as illustrated in Figure 2.1.

2.4.1 Spatial Aggregation

Spatial aggregation is the most common way to aggregate the data based on spatial boundary [Zha+20; Moo+19]. Spatial aggregation can be performed based on polygons, administrative boundaries, and graphs. We discuss spatial aggregation types in detail.

Polygon. As discussed in Section 2.2.4, we can partition a geospatial region with user-defined polygon shapes. A grid (square or rectangle) is the most common way to represent a region [Moo+19; Li+22b]. In grid aggregation, data such as geospatial points or events are aggregated in a fixed-size grid cell, where the size of the grid cell is user-defined [Moo+19]. The commonly utilized approach for constructing grids is geohash. Geohash transforms a geographic location into alphanumeric strings [DFD21]. The longer the geohash length, the finer the resolution, resulting in smaller grid cells. For instance, geohash with lengths of five and six correspond to approximate grids of sizes 4.89km $\times$ 4.89km and 1.22km $\times$ 0.61km, respectively. The grid-based spatial aggregation comes up with challenges. The grid-based aggregation does not follow the underlying distribution of spatial data. For instance, dense traffic accident events in the city center may lie in multiple grid cells. Hence, some grid cells may get fewer traffic accident events than others, leading to an uneven distribution of events.

Administrative. An administrative region can be defined with census tracts or street segments [Zha+20]. Figure 2.3a illustrates the division of the North Rhine-Westphalia (a federal state of Germany) based on administrative districts and aggregates traffic accident events in each district. Such data aggregation helps city administrators make informed decisions, such as allocating more resources in densely populated areas. The effectiveness of this aggregation is hindered by the varying shapes and sizes of administrative boundaries adopted by city administrations. For instance, the Manhattan region (in the United States of America (USA)) is subdivided into 180 administrative districts based on street networks [Zha+20]. In contrast, another division of the Manhattan region is based on census blocks leading to 270 administrative districts [Li+24; ZLC23]. As a result, transitioning between different types of boundary configurations requires reprocessing the data, increasing computational demands.

Graph. In graph aggregation, the geospatial and mobility data is aggregated in graphs. A graph is typically represented as $G(V, E)$, where V is the set of nodes or vertices and E is the set of edges. For instance, the road network can be expressed as a graph, in which each vertex represents a road segment, and edges represent the connection between the road segments [SHD23]. The geospatial points, events, and trajectory data along the road segments can be aggregated to each node or road segment for several tasks, such as speed prediction and traffic forecasting.

2.4.2 Temporal Aggregation

In temporal aggregation, spatial granularity is fixed. Temporal aggregation aggregates the data based on timestamps for a particular region. An example of temporal aggregation is analyzing taxi ride data to understand peak hours of activities in a city center [Liu+12]. By aggregating the data into different time intervals, such as morning rush hour, afternoon, evening, and late night, urban planners can identify when taxi demand is high in the city center. These intervals can vary from application to application.

2.4.3 Spatio-temporal Aggregation

The geospatial and mobility data is aggregated over space and time in spatio-temporal aggregation, as illustrated in Figure 2.3b. At distinct time intervals, this data can be aggregated over spatial representations, such as grids, graphs, and administrative districts. For instance, aggregating spatial events for each 5km $\times$ 5km grid cell in each 15-minute interval in a given region [Moo+19]. This spatio-temporal aggregation allows a more fine-grained analysis of spatial and temporal patterns than the aggregations discussed above. However, data sparsity remains a challenge due to fixed and uniform aggregations. An adaptive aggregation method needs to be developed to address the data sparsity challenge.

2.4.4 Adaptive Aggregation

The previously presented aggregation methods do not capture the underlying distribution of the data. For instance, the geospatial spread of POIs does not conform to fixed boundaries such as grids or administrative boundaries. Similarly, traffic accident events at a specific junction can be partitioned across multiple grid cells. The above examples highlight the need for an adaptive geospatial aggregation technique

to handle such problems. This thesis addresses the data sparsity challenge for traffic accident event data by proposing a novel adaptive clustering method for accident prediction [DFD21].

2.5 Predictive Model Development

In predictive model development, we first create the representation required by the learning models, then input these representations to the predictive models and perform prediction tasks. In the following sections, we discuss the whole process in more detail.

2.5.1 Data Representation for Learning Models

Effective data representation is crucial for the model learning process to make accurate predictions. Data representation can vary depending on the data type and the model structure. The most common data representations are feature vectors, matrices, graphs, and latent representations that we have adopted in the thesis.

Feature vectors are the most common representation for learning models. A feature vector is a list of numerical values that describe an instance of data [VR04]. Each vector dimension corresponds to a specific feature, such as a raw measurement, an engineered feature, or a categorical variable converted to a numeric form. Given a dataset with n features, each data instance x_i can be represented as a n-dimensional feature vector:

$$x_i = [x_{i1}, x_{i2}, \ldots, x_{in}],$$

where x_i is the feature vector for the i-th instance. x_{ij} represents the j-th feature of the i-th instance. For instance, consider a feature vector $x_i = [2, 40, "rainy"]$ representing features for a specific geospatial region i. Here, the value 2 represents the number of traffic accidents, 40 indicates the number of POIs, and "rainy" is a categorical value describing the weather in the geospatial region i. To prepare this data for predictive models, the numerical values (2 and 40) are scaled between 0 and 1, and the categorical value ("rainy") is transformed into a numerical form utilizing techniques like one-hot encoding. Scaling the numerical values confirms that all features contribute equally to the analysis, preventing features with larger ranges (like the number of POIs) from dominating those with smaller ranges (like the number of traffic accidents) [SS20]. The one-hot encoding method converts categorical features, like "rainy," into a numerical form, allowing the model to distinguish between different categories (e.g., "rainy", "sunny", and "cloudy"). In this thesis, we create feature vectors based on categorical and numerical features from the data sources such as OSM [DFD21; DYD24].

Matrices are utilized when the data involves relationships between entities. Each matrix element can represent a relationship between two entities, such as correlations. A matrix M can be represented with a $m \times n$ matrix,

$$M = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{bmatrix},$$

where m and n are the number of rows and columns, respectively. For instance, in graphs, the relation between two entities can be represented with an adjacency matrix. Furthermore, in image processing, an image is often represented as a matrix, with each element corresponding to a pixel value. In this thesis, we segment spatial maps from OSM as images, transform each spatial map image as a matrix, and utilize the matrix representation for learning region representation [DYD24].

Graphs are represented as $G(V, E)$, where V is the set of nodes or vertices and E is the set of edges, as discussed in Section 2.4.1. The graph vertices can be associated with feature vectors. The connection between the vertices can be expressed with an adjacency matrix A, where A is a $|V| \times |V|$ matrix,

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between vertex } i \text{ and vertex } j \\ 0 & \text{otherwise,} \end{cases}$$

where 1 and 0 are the edge weights. For instance, edge weights with value one can indicate that a geospatial region i is spatially connected with a geospatial region j. The edge weights can also be calculated based on the similarity between the feature vectors of nodes. In this thesis, we construct graphs from hexagonal grid cells, called grid graphs. In grid graphs, the grid cells act as vertices, and the connection between the grid cells is based on the cosine similarity between the feature vectors of the nodes [DYD24].

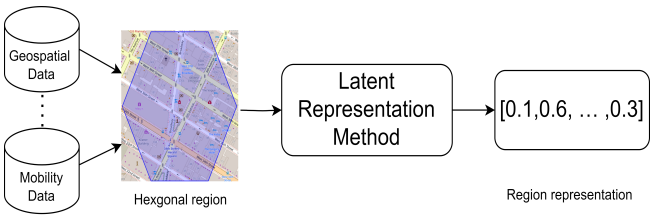


FIGURE 2.4: Geospatial region representation. Map data: ©OpenStreetMap contributors, ODbL

Latent representation refers to a compressed and meaningful data abstraction, represented in low-dimensional vectors, capturing essential features and underlying patterns [CG23]. Latent representation reduces storage requirements by transforming data from multi-modalities into a lower-dimensional space while preserving important information [She+23]. In the geospatial domain, the data from multi-modalities, such as geospatial and mobility data, is utilized to create the latent representation of the geospatial regions, as illustrated in Figure 2.4. This thesis focuses on creating effective and adaptive latent representations for geospatial regions. These effective representations retain the semantics for various downstream tasks and can be flexibly aggregated to any region of interest [DYD24].

2.5.2 Types of Models

This section explores predictive models that leverage machine learning and deep learning techniques for geospatial and mobility data.

Machine Learning. Machine learning (ML) models are algorithms that learn patterns from historical data and make predictions on unseen data [Sar21]. One of the main advantages of utilizing ML models is that they do not require large amounts of data to learn the patterns [JZH21]. Broadly, there are two types of learning in

ML: supervised ML and unsupervised ML [Sar21]. In supervised ML, the algorithm is trained on a labeled dataset, where each input data point is paired with a corresponding target label. Here, the input data may belong to one of the representations discussed in the previous section, e.g., feature vectors. During training, the algorithm adjusts the parameters to minimize the difference between the predictions and the actual labels provided in the training data. Common tasks in supervised learning include regression and classification. A regression task predicts a numerical value, while a classification task categorizes input data into different categories. Examples of supervised learning algorithms include linear regression, decision trees, gradient-boosting classifiers, support vector machines (SVM), and logistic regression. In the mobility domain, random forests and logistic regression models are commonly utilized in prediction tasks such as speed and traffic accident prediction [Bra+19; CC20]. In this thesis, we utilize logistic regression [Cox58] and gradient-boosting classifier [Fri01] as baseline methods for the traffic accident prediction task [DFD21]. Logistic regression is a statistical model for binary classification. It estimates the probability of an event occurring based on input features by fitting data to a logistic curve [Cox58]. On the other hand, a gradient-boosting classifier is an ensemble method. It builds multiple decision trees and combines the predictions of the decision trees sequentially [Fri01].

In unsupervised ML, the algorithm is given unlabeled data without specific target outputs. Unsupervised learning aims to learn the underlying patterns within the data without supervision. The common tasks in unsupervised learning include clustering and dimensionality reduction. In this thesis, we leverage clustering methods, such as KMeans [Mac67], DBSCAN [Est+96], HDBSCAN [CMS13], and self-organizing map (SOM) [Koh95] to group geospatial coordinates based on spatial proximity [DFD21]. KMeans is an iterative algorithm for partitioning the data into a 'K' number of clusters [Mac67]. In contrast, DBSCAN is a density-based clustering algorithm, requiring two input parameters: a minimum number of points and a radius-defining the neighborhood of each data point [Est+96]. DBSCAN can detect clusters of different shapes and sizes and distinguish outliers in the data compared to KMeans [KJ16]. Similarly, HDBSCAN builds on the principles of DBSCAN, which requires only one input parameter, i.e., a minimum number of points to form a cluster [CMS13]. Finally, SOM is an unsupervised ML method that maps high-dimensional data onto a 2D grid, where similar data points are grouped into neighboring nodes [Koh95].

Deep Learning. With the increasing volume of data, deep neural network-based models have demonstrated superior predictive abilities compared to classical machine learning models [DTV19]. Deep learning refers to Deep Neural Networks (DNN), structured with multiple layers of neurons that enhance their capacity for expression and performance [Cha+20]. We discuss three DNN models relevant to the thesis: Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Graph Neural Networks (GNNs) for processing sequential, grid, and graph data, respectively.

Recurrent Neural Networks (RNNs) [RHW86] models are deep learning models tailored for sequential data, such as trajectory, text, and temporal data. RNNs employ a recurrent mechanism to learn the sequential patterns [PMB13]. However, RNNs are subject to vanishing gradient problems. This occurs when the gradient diminishes substantially over time, which makes learning long-term dependencies challenging. Variants of RNNs such as Long-Short-Term Memory (LSTM) [HS97]

and Gated Recurrent Unit (GRU) [BCB15] have been introduced to address the gradient vanishing problem. These variants are designed to better retain the information over longer sequences. In the mobility domain, RNN models have applications in mobility-related tasks, including predicting road segment speeds within fixed time intervals [XL23] and predicting traffic accidents within a specified grid over the next hour [Moo+19]. This thesis utilizes GRU for the traffic accident prediction task [DFD21]. The advantages of GRU are that GRU has a smaller number of model parameters and is computationally less expensive than LSTM [Chu+14].

Convolutional Neural Networks (CNNs) [LeC+98] learn features from input data with grid patterns, such as images [Yam+18] and spatial grids. They are widely utilized in image and video recognition tasks [Li+22a]. In the mobility domain, CNNs are also utilized to learn the latent representation of segmented spatial maps [Var+19]. CNNs can recognize spatial patterns and structures in segmented spatial maps, which makes CNNs useful for applications such as land use classification [Ver+21]. In this thesis, we customize a variant of the CNN model, the EfficientNet model [TL19], to learn the representation of OSM map images for region embeddings [DYD24].

Graph Neural Networks (GNNs) [Sca+09] are applied to the graph-structured data consisting of nodes and edges. GNNs are neural architectures designed to capture the relationships within graphs by exchanging messages between the nodes [Zho+20]. Common variants of GNNs include Graph Convolutional Networks (GCN) [KW17] and Graph Attention Networks (GAT) [Vel+18]. The main difference between GCN and GAT is that these models employ different mechanisms for aggregating information in the neighborhood to capture graph dependencies. GCNs aggregate information from neighboring nodes, e.g., by taking a weighted sum of the features. Meanwhile, the GAT utilizes attention mechanisms to weigh the importance of neighboring nodes dynamically during information aggregation. GAT performs better than GCN due to attention mechanisms [Vel+18]. In this thesis, we leverage GAT to learn the latent representation of geospatial regions [DYD24].

2.5.3 Predictive Tasks

This section discusses predictive tasks such as regression, classification, and clustering. We also discuss different methods to evaluate these tasks.

Regression. A regression task aims to predict a continuous value based on input features fed to the model. In this thesis, we evaluate the regression models utilizing metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2). In the evaluation settings, if n is the total number of observations, o_i represents the actual output value for the i-th observation, and $\hat{o}_i$ represents the predicted value for the i-th observation in the data, then the metrics MAE, RMSE, and R^2 are defined as follows:

- **MAE** measures the average absolute difference between actual and predicted values,

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{o}_i - o_i|.$$

- **RMSE** is determined by computing the square root of the average of the squared differences between actual and predicted values,

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{o}_i - o_i)^2}.$$

- $\boldsymbol{R^2}$ is also known as the coefficient of determination. It calculates how well the regression model aligns with the data and ranges from 0 to 1, where a value close to 1 indicates a better fit of the model to the data,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{o}_i - \hat{o}_i)^2}{\sum_{i=1}^{n}(o_i - \bar{o}_i)^2},$$

  where $\bar{o}_i$ is the average of the actual output values.

In this thesis, we utilize MAE, RMSE, and R^2 for regression tasks such as crime count and check-in count prediction [DYD24].

Classification. The objective of a classification task is to assign input data into predefined categories or classes. Classification tasks are usually evaluated utilizing accuracy, precision, recall, and F1-score metrics. To understand the classification metrics better, we rely on a confusion matrix, as illustrated in Table 2.1. In a binary classification problem, the confusion matrix is a 2×2 table with four cells, i.e., true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). True positives (TP) are the positive tuples correctly labeled by the classifier. False positives (FP) are the negative tuples incorrectly labeled as positive. True negatives (TN) are the negative tuples correctly labeled by the classifier. False negatives (FN) are the positive tuples mislabeled as negative. Next, the classification metrics are defined based on the confusion matrix.

TABLE 2.1: Confusion matrix

		Actual Label	
		Positive	Negative
Predicted Label	Positive	TP	FP
	Negative	FN	TN

- **Accuracy** is defined as the proportion of correctly classified instances among all instances,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

- **Precision** represents the proportion of correctly predicted positive instances out of all instances predicted as positive,

$$Precision = \frac{TP}{TP + FP}.$$

- **Recall**, also known as sensitivity or true positive rate, indicates the proportion of correctly predicted positive instances out of all actual positive instances,

$$Recall = \frac{TP}{TP + FN}.$$

- **F1-score** is the harmonic mean of precision and recall,

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}.$$

In this thesis, we utilize the F1-score as a metric in the traffic accident prediction task [DFD21].

Clustering. In clustering, a dataset is partitioned into groups or clusters to maximize the intra-cluster similarity and minimize the inter-cluster similarity. The evaluation metrics for clustering are divided into two parts: 1) when actual labels are present and 2) when actual labels are absent. In this thesis, we look into the first type, i.e., when actual labels are present, and utilize evaluation metrics such as Adjusted Rand Index (ARI) [HA85] and Normalized Mutual Information (NMI) [LFK09].

- **Adjusted Rand Index (ARI)** measures the similarity between the actual label and the clustering result obtained by the algorithm. It yields a score between -1 and 1, where 1 indicates perfect cluster similarity.

- **Normalized Mutual Information (NMI)** measures the mutual dependence between the true and predicted clustering, normalized between 0 and 1. NMI values close to 1 indicate strong agreement between the clustering.

When actual labels are absent, the metrics for assessing cluster quality include intra-cluster variability and the Silhouette coefficient. In this thesis, we utilize NMI and ARI metrics for land use classification tasks when evaluating the latent representations of geospatial regions [DYD24].

Chapter 3

Literature Review

This chapter overviews state-of-the-art methods, focusing on event predictions, region representations, and watermarking techniques. First, we review several state-of-the-art methods that rely on fixed geospatial aggregations for traffic accident prediction tasks. Next, we discuss the related works on the latent representation of geospatial regions. Finally, we discuss state-of-the-art watermarking methods from media and mobility domains.

3.1 Geospatial Aggregations for Accident Event Predictions

Traffic accident prediction tasks are crucial in urban safety and planning, providing valuable insights for addressing real-world problems. Due to the data sparsity, accurate prediction of accident events often relies on geospatial aggregations. The event prediction tasks often involve aggregating data from multiple sources over grids of different sizes and administrative districts with varying levels of temporal granularity [YZY18; Moo+19]. For instance, Moosavi et al. developed a deep accident prediction (DAP) model [Moo+19] that aggregates data into 5km $\times$ 5km grids to predict accident occurrences in 15-minute intervals. They enriched each grid with data from several sources, such as weather information, POIs count, and temporal features. Similarly, Yuan et al. [YZY18] considered a similar geospatial granularity but extended the temporal granularity to one day for predicting traffic accident counts. They introduced Hetero-ConvLSTM, a deep-learning method that integrates data such as road conditions, weather, traffic volume, and satellite images for each grid. In addition, some studies, like Chen et al. [Che+16], reduced the geospatial granularity to 500m $\times$ 500m, demonstrating the variation in grid sizes for the traffic accident prediction task. They proposed a stacked denoise autoencoder (SdAE) approach incorporating historical traffic accident data and human mobility patterns. Recent work on traffic accident prediction considered hexagonal grids instead of square or rectangular grids [Mon+23]. They developed crashFormer, a deep learning method that leverages OSM map images, historical traffic accidents, and weather information for traffic accident prediction.

Existing approaches for traffic accident prediction tasks typically rely on fixed grids of arbitrary sizes, leading to an uneven distribution of accident events in each grid cell. This thesis proposes a novel adaptive clustering method for accident prediction (*ACAP*) to address the challenge of data sparsity [DFD21]. *ACAP* learns the underlying distribution of traffic accident events through adaptive clustering and performs accident prediction on dynamically determined clusters. Our experimental results demonstrate the effectiveness of the *ACAP* approach in the traffic accident prediction task.

3.2 Latent Representation of Geospatial Regions

Multimodal data is commonly utilized to create latent representations for geospatial regions. The geospatial region representations have applications in several tasks, such as land use classification, region popularity prediction, and crime forecasting. However, integrating data from multimodal sources is challenging due to data heterogeneity, resulting in ineffective latent representations. In addition, the geospatial regions can vary in shape and size depending on the user regions of interest. The state-of-the-art methods construct region representations from multimodal sources, which are ineffective and do not align with the user ROIs. For instance, Zhang et al. [Zha+20] presented a multi-view graph representation approach (MVURE), which integrated multiple modalities such as mobility patterns, POI data, and location-based social network check-in data to embed the administrative districts. Similarly, Fu et al. [Fu+19] developed a multi-view POI-POI network and utilized human mobility data to create the region representation for administrative regions. Zho et al. [Zho+23] build a heterogeneous region graph with human mobility and POI data (HREP) to create the region representation for administrative districts.

In contrast, some state-of-the-art methods have focused on a single modality for region representation learning. For instance, Wu et al. [Wu+22] focused on leveraging mobility data and developed a multigraph fusion network (MGFN) for embedding fixed-sized regions. Similarly, Li et al. [Li+23] presented a contrastive learning-based method (RegionDCL) based on OSM building data for creating a latent representation of regions divided based on street segments. Unlike creating region representation for fixed administrative regions, Woźniak et al. [WS21] developed the region representation for the hexagonal grid cell (Hex2Vec). They incorporated POI data from OSM for each grid cell and applied the continuous bag of words (CBOW) method to generate the region representation.

In addition, existing methods have considered different configurations for the administrative regions, as discussed in Section 2.4.1. For instance, while most state-of-the-art methods generate region representations for Manhattan City with 180 administrative regions [Zho+23; Zha+20], some approaches have also considered 270 regions [Li+24; ZLC23]. As a result, transitioning between different types of region configurations requires retraining the region representation learning model.

The region representations developed by the state-of-the-art methods are ineffective and are not adaptive to the user ROIs. This thesis proposes a novel approach called *MAGRE* that creates effective and adaptive latent representations for geospatial regions [DYD24]. Our experimental results demonstrate the effectiveness and adaptability of *MAGRE* embeddings in several downstream tasks.

3.3 Watermarking GPS Trajectories

As discussed in Section 2.3, watermarking is a technique to embed the provenance information into the data to enhance traceability. Most of the research related to watermarking is carried out in the media domains such as audio, image, and videos [HF24; Luo+23]. In the audio domain, El-Wahab et al. [El-+21] and Naqash et al. [NMP24] utilized Empirical Mode Decomposition (EMD) to break down the audio signal into multiple Intrinsic Mode Functions (IMFs) and embedded the watermark vector into one of the IMFs. K. et al. [KSD11] introduced a blind audio watermarking approach, employing Singular Value Decomposition (SVD) and Quantization

Index Modulation (QIM) methods for both watermark embedding and verification. In the image domain, Hosseini et al. proposed a blind digital image watermarking method that combines Discrete Cosine Transform (DCT), Principal Component Analysis (PCA), and Discrete Wavelet Transform (DWT) methods [HF24] for watermarking images. In the video domain, Luo et al. [Luo+23] employed an adversarial training method to insert the watermark in the video.

The mobility domain, particularly GPS trajectories, remains largely unexplored in watermarking. Jin et al. [Jin+05] introduced a blind watermarking technique that embeds watermarks into GPS coordinates based on the trajectory shape. This method has limitations, particularly its ineffectiveness in scenarios with consecutive similar coordinates, such as stops within trajectory data. In contrast, the TrajGuard approach, a state-of-the-art method for watermarking GPS trajectories, employs a geometric transformation [Pan+19]. This technique partitions trajectories into sub-trajectories and embeds the watermark utilizing centroid distance.

The state-of-the-art watermarking methods in the mobility domain are neither robust nor effective. In addition, the utility of watermarked trajectories has not been studied by the existing methods. In this thesis, we propose an effective, robust, and utility-preserving watermarking approach called *W-Trace* that embeds a watermark into the GPS trajectory based on the Fast Fourier Transform (FFT) [Dad+22; Dad+24]. *W-Trace* is a non-blind method and embeds more watermarking information than state-of-the-art methods. Through experiments, we demonstrate that our proposed *W-Trace* approach is robust, effective, and preserves utility for downstream applications.

Chapter 4

An Adaptive Clustering Approach for Accident Prediction

Publication Details
Rajjat Dadwal, Thorben Funke, and Elena Demidova.
"An Adaptive Clustering Approach for Accident Prediction".
In Proceedings of The 24th IEEE International Intelligent Transportation Systems Conference, ITSC 2021. IEEE, 2021, pages 1405-1411.
DOI: 10.1109/ITSC48978.2021.9564564

This chapter addresses the first research question regarding data *sparsity* in traffic accident event data. To tackle this challenge, we propose a novel adaptive clustering method for accident prediction (*ACAP*). *ACAP* addresses the limitations of data sparsity posed by the traditional aggregation methods and enhances the accident prediction performance.

4.1 Introduction

Predicting accident events is a crucial task in the mobility domain, enabling urban safety and planning. However, prediction of traffic accident events is a challenging task. First, traffic accident events are scattered across diverse geographic locations and recorded at a point-level granularity. This fine granularity often leads to a scarcity of data, which makes predicting accident events at the point level challenging. Existing traffic accident prediction approaches utilize fixed geospatial aggregations, such as administrative districts or fixed grids, to handle the data sparsity challenge. However, these predefined aggregations do not accurately capture the underlying distribution of traffic accident events. For instance, traffic accident events within a specific region could be distributed across multiple grid cells, resulting in only a few events in each cell. Second, current accident prediction models often rely on data from US cities [Moo+19], which have a grid-like layout [Boe19]. This contrasts with cities in Europe, which have more circuitous spatial structures. As a result, the grid-based aggregation methods designed for US cities cannot be directly applied to European cities. Moreover, the state-of-the-art accident prediction methods emphasize feature selection and model architecture while overlooking the significance of geospatial aggregation.

This work proposes an adaptive clustering approach for accident prediction (*ACAP*), a novel method for inferring adaptive clusters from sparse accident data. We present a clustering-based grid-growing (GG) algorithm that identifies

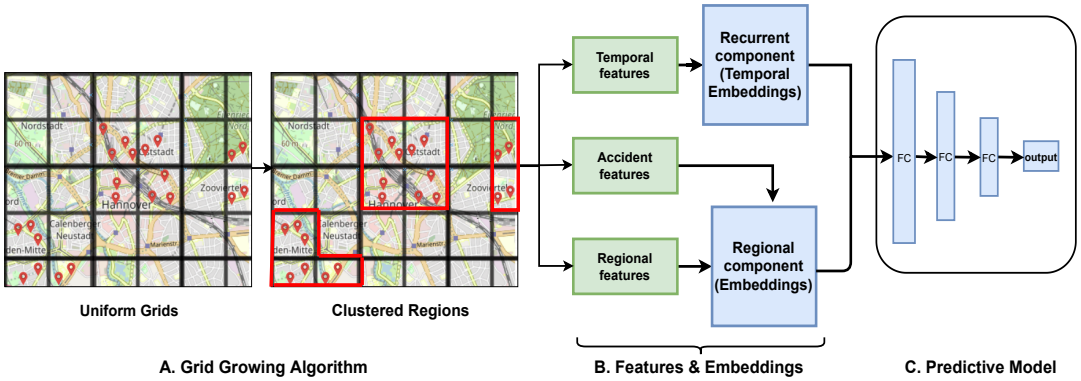


FIGURE 4.1: Architecture of the proposed *ACAP* approach [DFD21], ©2021 IEEE. Map data: ©OpenStreetMap contributors, ODbL

task-specific regions for accident prediction. In summary, our contributions are listed below:

- We propose a novel accident prediction approach, *ACAP*, which infers adaptive clusters formed by a grid-growing algorithm from sparse accident distributions.
- Our experiments demonstrate that *ACAP* outperforms the state-of-the-art methods by 2-3 percentage points on average regarding F1-score across three German cities.
- *ACAP* enhances accident predictions by two percent points in the F1-score compared to fixed grid aggregations 1km $\times$ 1km in more complex regions like city centers.

4.2 Definitions and Problem Formulation

Given the historical traffic accident observations, the goal is to create adaptive regions that follow the underlying distribution of accident events and predict binary accident events for these adaptive regions.

Definition 5 (Adaptive Accident Prediction) *Given a region formed through adaptive clustering of accident events, train a function* $\Phi \rightarrow \{0,1\}$ *such that* Φ *outputs '1' if an accident event is observed in the next period in the region and '0' otherwise [DFD21].*

4.3 Summary of the *ACAP* Approach

This section presents the *ACAP* approach. The overall architecture of the *ACAP* approach is illustrated in Figure 4.1. First, we introduce an adaptive clustering method designed to create clusters that align with the spatial distribution of accidents in Section 4.3.1. Next, in Section 4.3.2, *ACAP* incorporates data from different sources for each adaptive cluster and constructs temporal and geospatial feature embeddings. Finally, we present a predictive model of the *ACAP* approach in Section 4.3.3.

4.3.1 Adaptive Clustering with Grid Growing Algorithm

Existing traffic accident prediction methods often rely on uniform geospatial aggregation techniques, such as administrative districts or geohash, as prediction targets. These aggregations, often constrained by pre-aggregated data (e.g., for anonymization), result in coarse grids that do not represent the actual distribution of accident events. Additionally, existing studies evaluate the accident prediction methods on US-based datasets [Moo+19], where uniform grid structures align with typical city layouts [Boe19]. In contrast, European cities have more complex road layouts that deviate from a grid pattern. The challenges outlined above motivate us to perform adaptive clustering, generating geospatial aggregations that align more closely with the infrastructure and road layout of the target region.

We introduce an adaptive clustering method that accurately captures the spatial distribution of accident events. Adaptive clustering consists of two steps: i) grid construction and ii) grid growing, as illustrated in Figure 4.1a. In the grid construction step, a small-sized geohash of length seven is utilized to build grids of size 150m $\times$ 150m in a given region. In the grid growing step, we randomly choose a seed representing a grid cell with accidents and expand the area from the current seed by searching for accidents in the neighbor cells. When no accidents are found in adjacent cells, expansion terminates, and a cluster is assigned to the resulting region. Subsequently, another seed cell is randomly selected from the remaining accident-prone grid cells. This algorithm iterates until all accident-prone grid cells are assigned to a cluster.

4.3.2 Features & Embeddings

We enrich each adaptive cluster obtained from the grid-growing algorithm with temporal, accident, and regional features, as illustrated in Figure 4.1b.

- The temporal features include ten temporal features: weekends, weekdays, months, years, seasons, hours of the day, daylight, solar inclination, solar position, and solar elevation. We encode all the temporal features with the one-hot encoding technique.
- The accident features include the road conditions and accident type during the accident and are transformed into one-hot-encoded vectors.
- Regional features are the infrastructural characteristics of roads, such as amenities count, number of crossings, and number of junctions. Regional features are normalized to the range between 0 and 1.

To generate temporal embedding, temporal features are fed to the Gated Recurrent Network (GRU) [BCB15]. The static features, such as accident and regional features, are passed through a feed-forward neural network to obtain the embeddings.

4.3.3 Predictive Model

The predictive model of *ACAP* generates softmax probabilities for accidents and non-accidents, which we convert into binary labels: '1' for accidents and '0' for non-accidents, as illustrated in Figure 4.1c. The predictive model input comprises temporal, regional, and accident embeddings. *ACAP* model processes these embeddings through the neural network layers with decreasing dimensionality, applying

the rectified linear unit (ReLU) activation function in the first three layers and the softmax function in the last layer to classify accident events. The predictive model is optimized utilizing categorical cross-entropy as the loss function.

4.4 Evaluation

As the first part of the evaluation, we evaluate the impact of geospatial clustering methods on accident prediction. We compare the grid-growing approach against K-means, DBSCAN, HDBSCAN, and self-organizing map (SOM) in the Hannover region for accident prediction. The grid-growing method outperforms all the clustering-based baseline methods by at least four percent points. To examine the general performance, we evaluate *ACAP* with various spatial aggregations and four prediction baseline methods: gradient boosting classifier (GBC), logistic regression (LR), deep neural network (DNN), and deep accident prediction (DAP) model [Moo+19]. In particular, we select four spatial aggregations: GG, SOM, 1km $\times$ 1km, and 5km $\times$ 5km. We consider three German cities: Hannover, Munich, and Nuremberg. In all the geospatial aggregations and all cities, *ACAP* approach obtains the highest F1-score in the accident prediction. We also assess the effectiveness of the grid-growing algorithm in urban areas, ranging from the Hannover city center to the larger Hannover region. In the inner-city center, *ACAP* with grid growing outperforms the uniform grids (1km $\times$ 1km) by two percent points. In the end, we examine the significance of the three feature groups, namely regional, temporal, and accident features. We observed that temporal and regional features are crucial for accident prediction.

4.5 Discussion

In this chapter, we summarized *ACAP*, a novel accident prediction method based on a grid-growing approach to tackle the data sparsity challenge presented in Chapter 1. The existing approaches (e.g., [Moo+19]) primarily focused on feature selection and predictive model architecture and neglected the importance of geospatial aggregation in accident prediction. For geospatial aggregations, these methods often relied on uniform geospatial aggregation or administrative districts that do not align with the spatial distribution of the accident event data and lead to data sparsity.

In the *ACAP* approach, we included geospatial aggregation as an essential factor in modeling alongside feature selection and model architecture to tackle the data sparsity challenge in accident prediction tasks. We proposed a novel adaptive clustering method for accident prediction, which generates adaptive and task-specific regions for accident prediction. We performed accident predictions on these task-specific regions obtained through the proposed grid-growing method. For the predictive model, we utilized a neural network that combines temporal and static regional feature embeddings and predicts accident events in these adaptive regions.

Experiments on real-world datasets demonstrated the effectiveness of the adaptive clustering approach on accident prediction tasks. On average, *ACAP* improved the F1-score by 2-3 percentage points compared to the best-performing baseline methods in three German cities. The grid-growing algorithm adapted dynamically to the accident patterns and enhanced the F1-score by four percentage points over clustering-based baseline methods. Our adaptive clustering method based on the grid-growing algorithm is aligned with the distribution of traffic accident events and

enhanced predictions compared to baseline methods. In future work, we can investigate the effectiveness of adaptive aggregation methods in other event prediction tasks, such as crime prediction.

4.6 Contributions

I contributed to the conceptualization of the adaptive clustering approach. I also developed a neural network-based approach for traffic accident prediction. In addition, I carried out the implementations, experiments, and evaluations for the *ACAP* approach. Lastly, I contributed to the manuscript's writing and review.

Chapter 5

A Multimodal and Multitask Approach for Adaptive Geospatial Region Embeddings

Publication Details
Rajjat Dadwal, Ran Yu, and Elena Demidova.
"A Multimodal and Multitask Approach for Adaptive Geospatial Region Embeddings".
In Proceedings of 28th Pacific-Asia Conference on Knowledge Discovery and Data Mining Conference, PAKDD 2024. Springer, 2024, pages 363-375.
DOI: 10.1007/978-981-97-2262-4_29

This chapter addresses the second research question regarding ineffective latent representations for geospatial regions. To tackle this challenge, we propose *MAGRE* – an effective and adaptive latent representation learning method for geospatial regions. The latent presentations generated by *MAGRE* can be aggregated to any regions of interest (ROIs).

5.1 Introduction

Geospatial region representation is crucial for capturing spatial relationships within and between regions. The region representations are beneficial in several applications, such as land use classification and predicting crime rates [Wu+22; Zho+23]. However, the representation developed by existing methods may not match the regions or tasks of user interest and is subject to different limitations. First, the traditional geospatial representation learning methods depend on fixed administrative boundaries, such as districts [Zha+20; Wu+22] and are not adaptive to the ROIs. Second, existing approaches incorporate satellite imagery [Xi+22] as multimodal contextual information and require substantial preprocessing effort. Finally, region representations are typically optimized for specific tasks [Wu+22] and are ineffective for unseen tasks.

This work proposes a novel method, *MAGRE*, for creating effective and adaptive representations for geospatial regions. *MAGRE* embeds smaller geospatial units (grid cells) and dynamically aggregates the representations into an ROI as needed. However, such an adaptive aggregation comes with challenges. The overall semantics of the ROI may differ from those of its constituent grid cells. Furthermore, integrating geospatial and mobility data for each grid cell from multimodal sources is

challenging due to heterogeneity, resulting in ineffective representations for geospatial regions. To address these challenges, we propose a multimodal and multitask approach that incorporates rich visual information and graph context. In particular, we first partition the region into multiple hexagonal grid cells, extract features for each grid cell from multi-modalities, and learn the grid cell representation through multitasking. In summary, our contributions are as follows:

- We propose *MAGRE*, an effective and adaptive region representation learning method that captures region semantics utilizing a multimodal and multitask approach. *MAGRE* can embed regions of varying shapes and sizes through effective aggregation.
- *MAGRE* incorporates data from multiple sources to construct region representation. We integrate visual information from map images into region representations, effectively capturing the context of urban regions. The feature analysis confirms the significance of OSM images across downstream tasks.
- *MAGRE* employs multitask learning to enhance the effectiveness of the region representations. Experimental results demonstrate that *MAGRE* outperforms state-of-the-art methods in several downstream tasks, such as crime rate and check-in count predictions.

5.2 Definitions and Problem Formulation

In this section, we introduce the relevant definitions and the problem statement for spatial region representations.

Definition 6 (Geospatial grid cell) *A geospatial grid cell, represented as g, refers to the smallest spatial unit defined by specific geometric boundaries. Each grid cell is associated with features belonging to various categories. The features of a specific grid cell g_i corresponding to a feature category f are expressed as a vector $\vec{h}^i_f$ [DYD24].*

We partition the given region into hexagonal geospatial grid cells. A feature category f can represent the mobility patterns or population count. The relationships between grid cells concerning a specific feature category f are modeled as a grid graph.

Definition 7 (Grid graph) *A grid graph is represented as $\mathcal{G}_f = (\mathcal{V}, \mathcal{E}, \mathcal{A}_f)$, where $\mathcal{V} = \{g_1, ..., g_n\}$ is the set of grid cells, and $\mathcal{E}$ is the set of edges capturing the connections between grid cells. $\mathcal{A}_f$ is the weighted adjacency matrix associated with the feature category f. $A^{ij}_f = sim(\vec{h}^i_f, \vec{h}^j_f)$, where $sim(\cdot)$ denotes the similarity function [DYD24].*

The similarity between grid cells can be computed using the cosine similarity of the feature vectors. To facilitate efficient representation of grid cells, we rely on embeddings.

Definition 8 (Grid cell embedding) *The embedding of a grid cell g_i is defined as $e_i = \phi(g_i)$, $e_i \in R^d$ is a d-dimensional dense vector representation of g_i. The embedding function $\phi(\cdot)$ captures semantic and contextual information of g_i [DYD24].*

This thesis addresses the challenge of ineffective latent representation for geospatial regions. These regions can be administrative districts or new business areas (ROI).

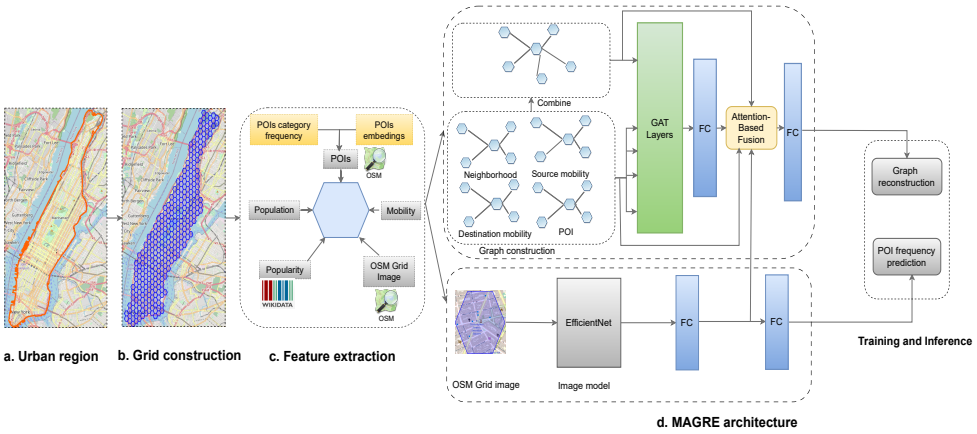


FIGURE 5.1: The overall architecture of the proposed *MAGRE* approach [DYD24], Copyright ©2024 Dadwal et al. Map data: ©OpenStreetMap contributors, ODbL

Definition 9 (Spatial region) *A spatial region is a geographic area defined by specific boundaries, denoted as r. A spatial region r can be represented by a set of spatial grid cells* $\{g_1, ..., g_n\}$ *it either contains or intersects with [DYD24].*

We aim to generate effective latent representations specific to any geospatial region derived from the representation of the spatial grid cells within that region.

Definition 10 (Spatial region representation) *For a given spatial region r, the geospatial representation* e_r *is obtained by aggregating the representations of the grid cells within the region* $e_r = \gamma(\{\phi(g_i)\})$*, where* $g_i \in r$ *and* $\gamma(\cdot)$ *is an aggregation function, such that* e_r *preserves the semantics of r [DYD24].*

5.3 Summary of the *MAGRE* Approach

This section presents the *MAGRE* approach. The overall architecture of the *MAGRE* approach is illustrated in Figure 5.1. In the proposed *MAGRE* approach, we first segment the geographic area into hexagonal grid cells and incorporate various features for each grid cell, as presented in Section 5.3.1. Then, in Section 5.3.2, we describe the proposed multimodal and multitask learning approach to learn the effective and adaptive latent representation for geospatial regions. Finally, we present the embedding aggregation method in Section 5.3.3.

5.3.1 Grid Construction and Feature Extraction

The initial data preprocessing step involves partitioning the entire graphic area into hexagonal grid cells, as illustrated in Figure 5.1b. Then, we extract several features for each grid cell from different data sources. Specifically, we represent each grid cell through feature vectors that depict counts of points of interest ($\vec{h}_{poi}$), mobility patterns ($\vec{h}_{mob}$), population and popularity counts ($\vec{h}_{pp}$). We also segment OSM images for each grid cell. These features are illustrated in Figure 5.1c.

5.3.2 *MAGRE* Model Architecture

This section presents *MAGRE* model architecture for creating effective and adaptive representations for geospatial regions, illustrated in Figure 5.1d. To design the objective functions, we consider two tasks: grid graph reconstruction and predicting POI frequency in each grid cell. To learn the effective latent representation from multimodal data, we employ an attention-based fusion, followed by training and inference.

For grid graph reconstruction, we first generate various graphs to capture semantic and spatial similarities among the grid cells. In particular, we construct a grid graph $\mathcal{G}_{poi}$ derived from POIs frequency ($\vec{h}_{poi}$), two grid graphs $\mathcal{G}_{src}$ and $\mathcal{G}_{dst}$ are generated based on mobility patterns, depicts source and destination graphs, respectively ($\vec{h}_{mob}$). In addition, a grid graph $\mathcal{G}_{nbh}$ is built to incorporate neighborhood information, capturing relationships between grids based on their geospatial proximity. We also construct a combined grid graph, represented as $\mathcal{G}_{cmb} = (\mathcal{V}, \mathcal{A}_{cmb})$, formed by averaging the adjacency matrices from all the individual grid graphs, such that $\mathcal{A}_{cmb} = \frac{1}{|f|}\sum_{i=1}^{|f|}\mathcal{A}_i$, where $f \in \{poi, nbh, src, dst\}$ [DYD24]. Averaging adjacency matrices identify grid cells with strong and consistent connections across modalities.

To derive meaningful representations from these grid graphs, graph attention network (GAT) [Vel+18] is employed. The GAT efficiently propagates information to neighboring grids within each graph, effectively updating grid representations. To predict POI frequency in each grid image, we customize the EfficientNet model [TL19], a variant of convolutional neural network (CNN) [LeC+98] to extract meaningful representations from grid images. Attention-based fusion techniques combine the representations from grid graphs and images. This fusion facilitates the propagation of knowledge across representations from different modalities. Next, the *MAGRE* model undergoes training and inference processes. The grid reconstruction task is trained unsupervised, and the reconstruction loss is computed utilizing mean squared error (MSE) loss. To train the grid images for the POI frequency prediction task, smooth L1 loss [Gir15] is employed. This loss function combines the benefits of both L1 and L2 losses, effectively managing outlier values. Overall, the loss function for the *MAGRE* model combines reconstruction loss and smooth L1 loss. During training, all embeddings and the model parameters are learned jointly through backpropagation.

5.3.3 Embedding Aggregation for Spatial Regions

After generating the representation for grid cells, the next step involves aggregating the grid cells representations for a specified region, e.g., ROIs. The embedding aggregation is achieved by summing the representation of grid cells within or intersecting with the ROIs. In the end, the aggregated representations are utilized for downstream tasks.

5.4 Evaluation

We evaluate the effectiveness and adaptiveness of the spatial region representation created by *MAGRE* on unseen tasks. Consistent with prior studies, we focus on the Manhattan City area. We conduct experiments on three distinct downstream

tasks: two regression tasks, predicting crime rates and check-in counts, and one classification task, land use classification. We compare *MAGRE* approach against several state-of-the-art baseline methods such as HREP [Zho+23], MG-FN [Wu+22], MVURE [Zha+20], Hex2Vec [WS21], RegionDCL [Li+23] and MV-PN [Fu+19]. In regression tasks, *MAGRE* exhibits superior performance compared to baseline methods, achieving lower MAE, RMSE, and higher R^2 scores. Regarding land use classification, *MAGRE* achieves the highest ARI score, outperforming the best baseline methods by 3.63%. These comprehensive evaluation results across all three tasks highlight the effectiveness of the latent representation generated from *MAGRE*.

To assess the influence of different features on model performance, we systematically eliminate one feature category at a time. Our results demonstrate that the best outcomes across all three tasks are achieved when all features are utilized, highlighting the effectiveness of the *MAGRE* approach in capturing region semantics. Notably, removing the OSM image significantly increases MAE and RMSE for regression tasks, alongside a decline in NMI and ARI for land use classification, demonstrating the importance of the OSM images. We also conduct a case study to demonstrate the adaptiveness of *MAGRE* representations by predicting crime rates in ROIs with varying sizes and shapes. *MAGRE* outperforms the selected baseline methods, i.e., HREP and MVURE, achieving a 63.61% reduction in MAE and a 52.02% reduction in RMSE compared to the best-performing baseline. These results highlight the adaptiveness of *MAGRE* latent representations in handling varying ROIs.

5.5 Discussion

In this chapter, we summarized *MAGRE*, an effective and adaptive approach for geospatial region representations to tackle the challenge of ineffective latent representations presented in Chapter 1. The region representations generated by existing methods are ineffective in capturing the semantics of the geospatial regions. Furthermore, these region representations are based on fixed boundaries and do not align with the user regions of interest. This misalignment between the ROIs and the region representation provided by the state-of-the-art methods results from the substantial limitations of the existing geospatial region representation approaches [DYD24].

We proposed a multimodal and multitasking approach to create effective and adaptive geospatial region representation. *MAGRE* constructed region representation by embedding smaller grid cells and dynamically aggregating the grid representations for a user region of interest. The aggregated adaptive region embeddings for ROIs effectively retained the semantic information, as confirmed by experiments on several downstream tasks.

Experimental results across three downstream applications demonstrated *MAGRE* superior performance over the state-of-the-art methods, confirming the effectiveness of multitasking and multimodal approach for urban region representation. In particular, our experimental results demonstrated that *MAGRE*'s representations outperform baseline methods, reducing root mean squared error by 25.73% for crime rate prediction and by 19.08% for check-in count prediction. In addition, the use case study on crime prediction task indicated the adaptiveness of *MAGRE*'s representations for different ROIs. In future work, we can fine-tune the grid cell embeddings for specific tasks to assess the effectiveness of the region embeddings.

5.6 Contributions

I contributed to the conceptualization of the adaptive geospatial region representation approach. In addition, I carried out the implementations, performed different experiments, and evaluated the *MAGRE* approach for its adaptiveness. Lastly, I contributed to the manuscript's writing and review.

Chapter 6

Towards Effective, Robust and Utility-preserving Watermarking of GPS Trajectories

Publication Details

- Rajjat Dadwal, Thorben Funke, Michael Nüsken, and Elena Demidova. "Towards effective, robust and utility-preserving watermarking of GPS trajectories." *Accepted for publication in ACM Transactions on Spatial Algorithms and Systems, TSAS, accepted on 03 October 2024.* DOI: 10.1145/3701558
- Rajjat Dadwal, Thorben Funke, Michael Nüsken, and Elena Demidova. "W-trace: robust and effective watermarking for GPS trajectories." *In Proceedings of the 30th International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2022. ACM, 2022, pages 77:1–77:4 (Short paper).* DOI: 10.1145/3557915.3561474

This chapter addresses the third research question regarding the *lack of traceability* in personal mobility data. To tackle this challenge, we propose a robust, effective, and utility-preserving watermarking approach for GPS trajectories.

6.1 Introduction

GPS trajectories are the most widely utilized mobility data for predictive tasks such as speed prediction, trajectory user linking, and next location prediction [CF22]. However, GPS trajectories often contain sensitive information such as visited locations, personal preferences, and home addresses. Sharing GPS trajectory data for different tasks, such as predictive model development, can raise concerns [Dad+22]. Applications dependent on such data need effective and robust methods to verify provenance and authenticity.

Digital watermarking embeds watermark information into noise-tolerant data, enabling verification of provenance and authenticity of data. However, the watermarking of GPS trajectories comes with challenges. The primary challenge is managing the tradeoff between effectiveness and robustness of watermarking while minimizing the impact on data utility. A watermark must embed enough information for verification and resist modification by adversaries while preserving the data utility of watermarked GPS trajectories for downstream applications. In addition, the

non-uniform sampling rates and positional inaccuracies of GPS trajectories make the trajectories vulnerable to modification attacks like point removal, addition, and resampling. Current watermarking methods either embed limited provenance information into the trajectories or lack robustness [Pan+19; Jin+05]. Furthermore, the utility of watermarked trajectories in downstream tasks remains largely unexplored. This work proposes a novel watermarking method, *W-Trace*, that is effective and robust to different attacks while preserving utility for downstream tasks.

In summary, our contributions are as follows:

- We propose *W-Trace*, a novel watermarking method for GPS trajectories that represents two-dimensional coordinates as complex numbers and employs Discrete Fourier Transform (DFT) to embed watermarks in the trajectory.
- We experimentally demonstrate that *W-Trace* is robust to several adversarial modifications, achieving an average recognition rate of 99% on two real-world datasets.
- *W-Trace* embeds more watermark information than state-of-the-art methods by dispersing the watermark throughout the trajectory, enhancing effectiveness and robustness.
- *W-Trace* minimizes trajectory modification to preserve essential characteristics and maintain utility for real-world applications, such as map matching and trajectory user linking.

6.2 Definitions and Problem Formulation

This section introduces the relevant definitions and the problem statement. According to the Definition 3, a GPS trajectory T consists of geospatial points organized chronologically and paired with their corresponding timestamps. These GPS trajectories are watermarked with a watermark embedding process.

Definition 11 (Watermark embedding) *For a GPS trajectory T, a watermark sequence W is inserted into T through an embedding function $EMB(\cdot)$,*

$$\tilde{T} = EMB(T, W),$$

where $\tilde{T}$ is the watermarked trajectory [Dad+24].

Watermark verification is a process of determining whether a given watermark sequence is inserted into the trajectory.

Definition 12 (Watermark verification) *Given an original trajectory T, a GPS trajectory $\tilde{T}$, and a watermark sequence W, the verification function*

$$VER(T, W, \tilde{T}, \theta_v) \rightarrow B, B \in \{true, false\}$$

evaluates whether the specified watermark sequence W is inserted into the trajectory $\tilde{T}$. θ_v are approach-specific verification parameters [Dad+24].

An adversary can alter the watermarked trajectory $\tilde{T}$ to destroy or remove the watermark. This alteration is considered an attack, denoted as $\hat{\tilde{T}} = AT(\tilde{T}, \theta)$ on the watermarked trajectory $\tilde{T}$, where θ represents the parameter for a specific attack, resulting in an attacked trajectory $\hat{\tilde{T}}$.

Definition 13 (Attack) *Given a watermarked GPS trajectory $\tilde{T} = EMB(T, W)$, an attack $\hat{\tilde{T}} = AT(\tilde{T}, \theta)$ aims to hinder the watermark verification process [Dad+24]:*

$$VER(T, W, \tilde{T}, \theta_v) \rightarrow B, VER(T, W, \hat{\tilde{T}}, \theta_v) \rightarrow B', B' \not\equiv B.$$

A watermarking approach is considered robust against an attack $AT(\cdot)$ if the watermark verification function $VER(\cdot)$ outputs the same result for both the attacked trajectory $\hat{\tilde{T}}$ and the watermarked trajectory $\tilde{T}$.

Definition 14 (Robust watermarking) *Given a watermarked GPS trajectory $\tilde{T}$, an attack $\hat{\tilde{T}} = AT(\tilde{T}, \theta)$ and a watermark verification function $VER(\cdot)$, the watermarking is considered robust against $AT(\cdot)$ if $VER(\cdot)$ outputs equivalent labels for $\tilde{T}$ and $\hat{\tilde{T}}$ [Dad+24]:*

$$VER(T, W, \tilde{T}, \theta_v) \rightarrow B, VER(T, W, \hat{\tilde{T}}, \theta_v) \rightarrow B', B' \equiv B.$$

Trajectory modifications, such as watermarking and attacks, can impact the utility of trajectory data in real-world applications like accident prediction and driving behavior profiling. These applications are referred to as predictive models, denoted by $M(\cdot)$.

Definition 15 (Predictive model) *Given a GPS trajectory T, a predictive model $M(\cdot)$ takes a trajectory T and parameters $param$ as inputs and outputs a label L [Dad+24], i.e.,*

$$M(T, param) \rightarrow L,$$

where L is application-specific and can represent different categories, such as traffic speed or accident probability.

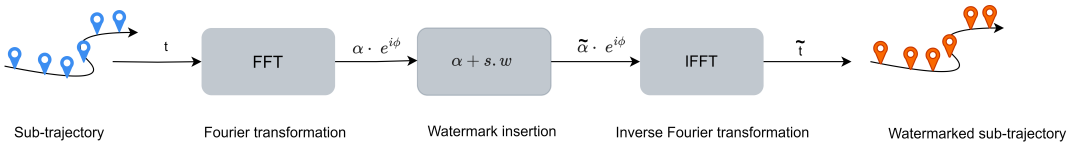
We define a trajectory modification as utility-preserving regarding $M(\cdot)$ if applying $M(\cdot)$ to the original and the modified trajectories output the same label.

Definition 16 (Utility-preserving modification) *Given a GPS trajectory T, and a predictive model $M(\cdot)$, the modification $\tilde{T} = MOD(T, ...)$ is considered utility-preserving regarding $M(\cdot)$ if applying $M(\cdot)$ to both $\tilde{T}$ and T output the same label [Dad+24]:*

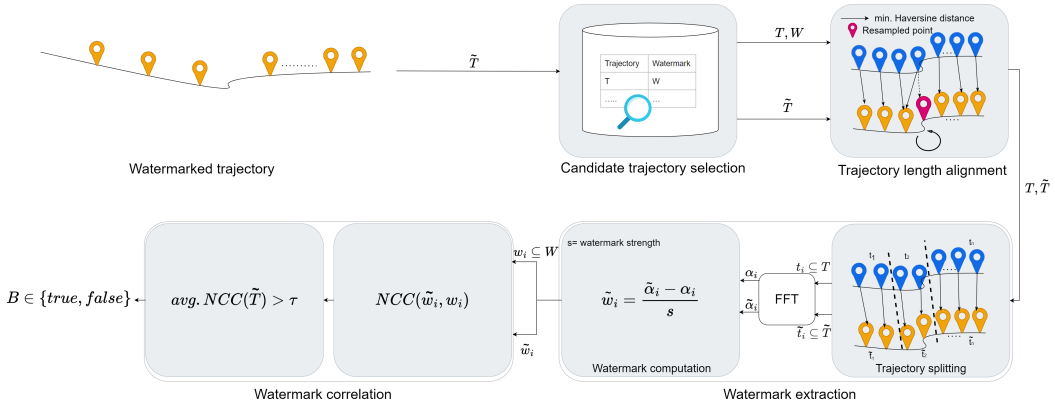
$$M(T, param) \rightarrow L, M(\tilde{T}, param) \rightarrow L', L' \equiv L.$$

6.3 Summary of the *W-Trace* Approach

In this thesis, we present *W-Trace*, an effective, robust, and utility-preserving watermarking approach for GPS trajectories. The overall architecture of the *W-Trace* is illustrated in Figure 6.1. The two main steps of the watermarking approach, as discussed in Section 2.3, are watermark embedding and watermark verification. First, in Section 6.3.1, we discuss the watermark embedding step that inserts a watermark into the GPS trajectories, as illustrated in Figure 6.1a. Then, in Section 6.3.2, we present the watermark verification step in which the watermark is first extracted from the modified trajectory and then verified, as illustrated in Fig. 6.1b.



A) Watermark embedding in *W-Trace*



B) Watermark verification in *W-Trace*

FIGURE 6.1: Overview of the *W-Trace* approach [Dad+24], ©2024 ACM

6.3.1 Watermark Embedding

The aim of the watermark embedding step is to insert a watermark into a given GPS trajectory. The watermark embedding process, denoted as $EMB(T, W)$ (in Definition 11), takes the trajectory $T = [(p_j, t_j)]$, where each point in p_j is defined as (lat_j, lon_j) and along with a watermark sequence W as an input and outputs the watermarked trajectory $\tilde{T}$. First, each GPS trajectory is partitioned into sub-trajectories $T = [t_1, ..., t_n]$ of equal length, matching the dimensions of the watermark vectors $w \in W$. The embedding function maps each GPS point $p_j = (lat_j, lon_j)$ within a sub-trajectory $t \in T$ to a complex number [Dad+24],

$$c_j = lat_j + ilon_j, \tag{6.1}$$

where i is the imaginary unit.

The benefit of utilizing complex numbers is that the watermark can be spread into both coordinates simultaneously. Next, a Discrete Fourier Transform (DFT) [Win78] is applied to each sub-trajectory. In particular, the Fast Fourier Transform (FFT) algorithm [Nus81] is employed to enhance computational efficiency. The FFT algorithm processes the list of positions $c = [c_j]_{1 \leq j \leq m}$ from the sub-trajectory, where these positions are represented as complex numbers. The resulting frequency components are represented in terms of amplitude α and phase angle ϕ [Dad+24]:

$$\alpha \cdot e^{i\phi} \leftarrow FFT(c). \tag{6.2}$$

The watermark $w \in W$ with strength $s \in (0, 1)$ is embedded into the amplitude α of the sub-trajectory t [Dad+24]:

$$\tilde{\alpha} = \alpha + s \cdot w. \tag{6.3}$$

In the *W-Trace* approach, the watermark w is represented as a vector, with each element randomly assigned a value of 0, 1, or -1, enabling a distinct watermark vector for each sub-trajectory. The next step is to apply an inverse FFT (IFFT) to obtain the watermarked trajectory:

$$\tilde{t} = (\tilde{a}, \tilde{b}) \leftarrow \ IFFT(\tilde{\alpha} \cdot e^{i\phi}), \tag{6.4}$$

where $\tilde{t}$ is a watermarked sub-trajectory [Dad+24]. All the watermarked sub-trajectories are concatenated into the watermarked trajectory $\tilde{T}$.

6.3.2 Watermark Verification

Watermark verification involves determining whether a specific watermark sequence W is embedded within a given trajectory $\tilde{T}$. As stated in Definition 12, the verification function is expressed as: $VER(T, W, \tilde{T}, \theta_v) \rightarrow B, B \in \{true, false\}$. Here, T represents an original trajectory, W denotes the watermark sequence, and $\tilde{T}$ is a GPS trajectory to be verified. In this context, θ_v corresponds to the watermark strength parameter s utilized during the watermark embedding process.

The watermarked GPS trajectories are vulnerable to adversarial modifications, also known as attacks. The attacks considered in this work have been previously explored in the literature within the domains of trajectory watermarking [Pan+19; Dad+22], cryptography [HPC10] and trajectory similarity measures [Su+20]. In particular, this work considers four types of attacks: noise additive attacks, length modification attacks, point replacement attacks, and hybrid attacks. These attacks can alter the watermarked trajectory, leading to a modified trajectory.

The watermark verification is performed to verify the watermark in the modified trajectory. The watermark verification assesses whether a given watermark sequence is embedded into a modified trajectory through a four-step process: candidate trajectory selection, trajectory length alignment, watermark extraction, and watermark correlation. In the first step, the verification begins by selecting the closest original trajectory as the candidate based on Haversine distance. Then, trajectory length alignment is performed by resampling to match the lengths of the candidate and modified trajectory. In the watermark extraction step, the watermark is extracted from the modified trajectory by utilizing the candidate trajectory. We extract the watermark by [Dad+24]:

$$\tilde{w} = \frac{\tilde{\alpha} - \alpha}{s}, \tag{6.5}$$

where s is the watermark strength, $\tilde{w} \in \tilde{W}$, and α is the amplitude of the candidate original trajectory T. Finally, Normalized Cross-Correlation (NCC) is employed to compute the correlation between original and extracted watermarks, as discussed in Section 2.3. The watermark verification is successful if the NCC score exceeds the acceptance threshold (τ). We adopt the acceptance threshold based on [Pan+19] ($\tau > 0.85$).

6.4 Evaluation

To evaluate the effectiveness and robustness of watermark verification, we employ several metrics commonly utilized in assessing watermarking approaches, including watermark recognition rate, false-positive rate, and embedding capacity. We utilize

two real-world trajectory datasets, the German and Porto datasets, each containing 1100 randomly selected trajectories of length 256. The utility of watermarked trajectories is assessed through two downstream tasks: map matching and neural network-based predictive task-trajectory user linking (TUL). We utilize the Jaccard similarity coefficient for map matching as the evaluation metric, while accuracy is used for the TUL task.

We compare the *W-Trace* approach to the baseline methods, including Intrinsic Mode Function (IMF) [El-+21], Singular Value Decomposition (SVD) [KSD11] from the audio domain, and TrajGuard [Pan+19] from the mobility domain. The proposed *W-Trace* approach demonstrates effectiveness and robustness against all the considered attacks, achieving an average recognition rate of 99% on the German and Porto datasets. Unlike the baseline methods, which demonstrate varying performance to the attacks in different datasets, *W-Trace* consistently performs well. Compared to the TrajGuard baseline, *W-Trace* has high embedding capacity. Additionally, *W-Trace* achieves a zero false-positive rate compared to the IMF watermarking method. Our results demonstrate that the trajectories watermarked by the *W-Trace* approach maintain the utility characteristics for map matching and trajectory user linking.

6.5 Discussion

In this chapter, we summarized *W-Trace*, an effective, robust, and utility-preserving watermarking approach for GPS trajectories to tackle the challenge of lack of traceability presented in Chapter 1. Existing watermarking methods are either ineffective, embedding only minimal provenance information [Pan+19] or lack robustness [Jin+05]. Moreover, previous research has not thoroughly studied the utility of the watermarked trajectories on downstream tasks [Pan+19; Jin+05].

To tackle these challenges, we presented an effective, robust, and utility-preserving method for watermarking GPS trajectories to enhance traceability. *W-Trace* utilized a Discrete Fourier Transform (DFT) to each sub-trajectory and embedded an imperceptible watermark into the Fourier descriptors. By embedding the watermark into each frequency component, *W-Trace* dispersed the watermark across the frequency components, allowing for more embedded information. Additionally, *W-Trace* controls the number of modifications introduced during watermark embedding to preserve the utility of the trajectories.

Experimental results across different datasets demonstrated *W-Trace*'s superior performance over the state-of-the-art methods. In particular, *W-Trace* achieved a watermark recognition rate of 99% on average on two real-world datasets, demonstrating the effectiveness and robustness of *W-Trace* against the modifications. The experimental results demonstrated that the GPS trajectories watermarked by the *W-Trace* approach preserved the utility for downstream applications such as map matching and predictive tasks such as trajectory user linking. Additionally, *W-Trace* embedded more watermark information into the GPS trajectory than the state-of-the-art methods. In future work, we can develop a domain-agnostic watermarking method for Internet of Things (IoT) data that is robust and utility-preserving, enabling watermarking applications across various domains.

6.6 Contributions

I contributed to conceptualizing the GPS watermarking approach, implemented the *W-Trace* method and baseline methods, performed experiments, and evaluated the watermarking approach for its effectiveness, robustness, and utility preservation. In the end, I contributed to the manuscript's writing and review.

Chapter 7

Discussion and Future Work

In this thesis, we identified and addressed challenges associated with geospatial and mobility data, such as data sparsity, ineffective region representations, and lack of traceability. These challenges serve as the foundation for developing novel methods in the mobility domain. We provide an in-depth discussion of our contributions and explore potential directions for future research.

7.1 Discussion of Contributions

In this thesis, we proposed novel adaptive methods for accident prediction and region latent representations. We also presented a robust, effective, and utility-preserving watermarking approach for GPS trajectories. Next, we discuss each contribution in detail.

In Chapter 4, we addressed the first research question regarding data sparsity in accident event data. Existing methods often rely on uniform geospatial aggregation or administrative districts for geospatial aggregations. These aggregations do not align with the spatial distribution of the accident event data and lead to data sparsity. To tackle this challenge, we proposed a novel adaptive clustering method for accident prediction called *ACAP*. As part of adaptive clustering (AC), the grid-growing algorithm created adaptive clusters based on the distribution of accident events. The features from multimodal data helped to enrich the context of each adaptive cluster for the accident prediction (AP) task. The evaluation results demonstrated the effectiveness of the *ACAP* approach on the accident prediction tasks. *ACAP* enhanced prediction performance by two percent points in the F1-score compared to fixed aggregations in the city center. The grid-growing algorithm outperformed the clustering-based methods by four percent points regarding the F1-score in the accident prediction task. The ablation study concluded that POIs and temporal features are crucial for accident prediction. In summary, the grid-growing algorithm addressed the challenges of data sparsity and improved accident prediction results, as demonstrated by different experiments on real-world datasets.

In Chapter 5, we addressed the second research question regarding ineffective latent representations for geospatial regions. The region representations generated by existing methods are ineffective at capturing the semantics of geospatial regions and are constrained by fixed boundaries that fail to align with user regions of interest. We proposed an effective and adaptive geospatial representation learning approach called *MAGRE*. *MAGRE* employed a multimodal and multitask learning approach with attention-based fusion, leading to effective and adaptive geospatial representations. The representations generated by *MAGRE* can be aggregated to any shape or size of ROIs. Experimental results on three downstream tasks highlighted

the effectiveness of the multimodal and multitasking approach. Specifically, the experimental findings demonstrated that *MAGRE* outperformed the state-of-the-art methods, resulting in a root mean squared error reduction of 25.73% and 19.08% for predicting crime rate and check-in count, respectively. Furthermore, the case study on crime prediction demonstrated the representations generated from *MAGRE* are adaptable across various ROIs. In summary, the geospatial region representations generated by *MAGRE* retain the semantics for several downstream tasks and can be adaptively aggregated to any ROIs.

Finally, in Chapter 6, we addressed the third research question regarding the lack of traceability in personal mobility data, particularly for GPS trajectories. The state-of-the-art watermarking methods for GPS trajectories are either ineffective, embedding only minimal provenance information [Pan+19] or lack robustness against the modifications [Jin+05]. We proposed a novel approach called *W-Trace*, which transforms GPS trajectories into Fourier descriptors utilizing the Fast Fourier Transform (FFT) and embeds imperceptible watermarks into these descriptors. *W-Trace* is robust to adversarial attacks compared to the state-of-the-art methods and achieved an average watermark recognition rate of around 99%. Our results demonstrated that the watermarked trajectories generated by the *W-Trace* approach retain the utility characteristics for downstream tasks. Furthermore, *W-Trace* incorporated more watermark information into the GPS trajectory than the state-of-the-art methods. In summary, the *W-Trace* approach is an effective, robust, and utility-preserving method that enhances traceability for GPS trajectories.

7.2 Open Research Directions

In this thesis, we presented novel approaches to tackle different challenges for geospatial and mobility data. Based on the observations and findings presented in this thesis, the following aspects can be explored in the future.

7.2.1 Adaptive Geospatial Aggregation

In the traffic accident prediction approach, *ACAP* performs adaptive clustering based on the spatial proximity of accident events. In future work, we can generate the latent representations of the traffic accident data and perform adaptive clustering on the latent representations of events. In addition, we can explore how our adaptive aggregation method (*ACAP*) can be applied to other event prediction tasks, such as crime prediction.

7.2.2 Adaptive Latent Representation

In the adaptive latent representation approach, *MAGRE* aggregates the embeddings for the user regions of interest from different grid cells and utilizes the aggregated embeddings for several predictive tasks. A potential direction for future research is to fine-tune the grid cell embeddings to specific tasks to explore the effectiveness of the embeddings. Furthermore, we can look into creating embeddings for POIs and adaptively aggregate the POIs embeddings for any ROIs.

7.2.3 Watermarking

In the *W-Trace* approach, we watermark GPS trajectories for authentication purposes. One possible direction for further research is to develop a domain-agnostic watermarking method for Internet of Things (IoT) data. The idea is to create a robust and utility-preserving watermarking method that can be applied to IoT data in different fields, such as the health domain. Furthermore, we can explore AI-based methods to authenticate the mobility data. Most AI-based research for data authentication is conducted in media domains, such as image, audio, and video [BP23; SBG23]. With the emerging applications of artificial intelligence in different tasks and domains, it is interesting to examine AI-based methods for data authentication in non-media domains and verify the robustness of AI methods.

Bibliography

[An+16] Shi An, Haiqiang Yang, Jian Wang, Na Cui, and Jianxun Cui. "Mining urban recurrent congestion evolution patterns from GPS-equipped vehicle mobility data". In: *Inf. Sci.* 373 (2016), pp. 515–526. DOI: 10.1016/J.INS.2016.06.033.

[APIa] OSM API. https://wiki.openstreetmap.org/wiki/API.

[APIb] Overpass API. https://wiki.openstreetmap.org/wiki/Overpass_API.

[BCB15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations, ICLR*. 2015.

[Boe19] Geoff Boeing. "Urban spatial order: street network orientation, configuration, and entropy". In: *Appl. Netw. Sci.* 4.1 (2019), 67:1–67:19. DOI: 10.1007/S41109-019-0189-1.

[BP23] Marta Bistron and Zbigniew Piotrowski. "Efficient Video Watermarking Algorithm Based on Convolutional Neural Networks with Entropy-Based Information Mapper". In: *Entropy* 25.2 (2023), p. 284. DOI: 10.3390/E25020284.

[Bra+19] Charalampos Bratsas, Kleanthis Koupidis, Josep-Maria Salanova, Konstantinos Giannakopoulos, Aristeidis Kaloudis, and Georgia Aifadopoulou. "A comparison of machine learning methods for the prediction of traffic speed in urban places". In: *Sustainability* 12.1 (2019), p. 142. DOI: 10.3390/su12010142.

[CC20] Mu-Ming Chen and Mu-Chen Chen. "Modeling Road Accident Severity with Comparisons of Logistic Regression, Decision Tree and Random Forest". In: *Inf.* 11.5 (2020), p. 270. DOI: 10.3390/INFO11050270.

[CF22] Ayele Gobezie Chekol and Marta Sintayehu Fufa. "A survey on next location prediction techniques, applications, and challenges". In: *EURASIP J. Wirel. Commun. Netw.* 2022.1 (2022), p. 29. DOI: 10.1186/S13638-022-02114-6.

[CG23] Shuangshuang Chen and Wei Guo. "Auto-Encoders in Deep Learning—A Review with New Perspectives". In: *Mathematics* 11.8 (2023). ISSN: 2227-7390. DOI: 10.3390/math11081777.

[Cha+20] Guillaume Chassagnon, Maria Vakalopolou, Nikos Paragios, and Marie-Pierre Revel. "Deep learning: definition and perspectives for thoracic imaging". In: *European radiology* 30 (2020), pp. 2021–2030. DOI: 10.1007/s00330-019-06564-3.

[Che+16] Quanjun Chen, Xuan Song, Harutoshi Yamada, and Ryosuke Shibasaki. "Learning Deep Representation from Big and Heterogeneous Data for Traffic Accident Inference". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 2016, pp. 338–344. DOI: 10.1609/AAAI.V30I1.10011.

[Chu+14] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: *CoRR* abs/1412.3555 (2014).

[CMS13] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. "Density-Based Clustering Based on Hierarchical Density Estimates". In: *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD, 2013*. Vol. 7819. Lecture Notes in Computer Science. Springer, 2013, pp. 160–172. DOI: 10.1007/978-3-642-37456-2_14.

[Cox58] David R Cox. "The regression analysis of binary sequences". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 20.2 (1958), pp. 215–232. DOI: 10.1111/j.2517-6161.1958.tb00292.x.

[Dad+22] Rajjat Dadwal, Thorben Funke, Michael Nüsken, and Elena Demidova. "W-trace: robust and effective watermarking for GPS trajectories". In: *Proceedings of the 30th International Conference on Advances in Geographic Information Systems, SIGSPATIAL*. ACM, 2022, 77:1–77:4. DOI: 10.1145/3557915.3561474.

[Dad+24] Rajjat Dadwal, Thorben Funke, Michael Nüsken, and Elena Demidova. "Towards effective, robust and utility-preserving watermarking of GPS trajectories". In: *ACM Trans. Spatial Algorithms Syst.* (2024). ISSN: 2374-0353. DOI: 10.1145/3701558.

[DFD21] Rajjat Dadwal, Thorben Funke, and Elena Demidova. "An Adaptive Clustering Approach for Accident Prediction". In: *24th IEEE International Intelligent Transportation Systems Conference, ITSC*. IEEE, 2021, pp. 1405–1411. DOI: 10.1109/ITSC48978.2021.9564564.

[DGL07] Michael John De Smith, Michael F Goodchild, and Paul Longley. *Geospatial analysis: a comprehensive guide to principles, techniques and software tools*. Troubador publishing ltd, 2007. DOI: 10.1111/j.1467-9671.2008.01122.x.

[DTV19] Loan N. N. Do, Neda Taherifar, and Hai Le Vu. "Survey of neural network-based models for short-term traffic state prediction". In: *WIREs Data Mining Knowl. Discov.* 9.1 (2019). DOI: 10.1002/WIDM.1285.

[DYD24] Rajjat Dadwal, Ran Yu, and Elena Demidova. "A Multimodal and Multitask Approach for Adaptive Geospatial Region Embeddings". In: *Advances in Knowledge Discovery and Data Mining - 28th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD*. Vol. 14649. Lecture Notes in Computer Science. Springer, 2024, pp. 363–375. DOI: 10.1007/978-981-97-2262-4_29.

[El-+21] Basant S. Abd El-Wahab, Heba Ali El-Khobby, Mustafa M. Abd-Elnaby, and Fathi E. Abd El-Samie. "Simultaneous speaker identification and watermarking". In: *Int. J. Speech Technol.* 24.1 (2021), pp. 205–218. DOI: 10.1007/S10772-019-09658-X.

[ESR] ESRI. *GIS Dictionary*. https://support.esri.com/en-us/gis-dictionary/polygon.

[Est+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*. AAAI Press, 1996, pp. 226–231.

[Far+16] Damien R Farine, Ariana Strandburg-Peshkin, Tanya Berger-Wolf, Brian Ziebart, Ivan Brugere, Jia Li, and Margaret C Crofoot. "Both nearest neighbours and long-term affiliates predict individual locations during collective movement in wild baboons". In: *Scientific reports* 6.1 (2016), p. 27704. DOI: 10.1038/srep27704.

[Fri01] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics* (2001), pp. 1189–1232. DOI: 10.1214/aos/1013203451.

[Fu+19] Yanjie Fu, Pengyang Wang, Jiadi Du, Le Wu, and Xiaolin Li. "Efficient Region Embedding with Multi-View Spatial Networks: A Perspective of Locality-Constrained Spatial Autocorrelations". In: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*. AAAI Press, 2019, pp. 906–913. DOI: 10.1609/AAAI.V33I01.3301906.

[Gir15] Ross B. Girshick. "Fast R-CNN". In: *2015 IEEE International Conference on Computer Vision, ICCV*. IEEE Computer Society, 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.

[HA85] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of classification* 2 (1985), pp. 193–218. DOI: 10.1007/BF01908075.

[Han+23] Sumin Han, Youngjun Park, Minji Lee, Jisun An, and Dongman Lee. "Enhancing Spatio-temporal Traffic Prediction through Urban Human Activity Analysis". In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM*. ACM, 2023, pp. 689–698. DOI: 10.1145/3583780.3614867.

[HF24] S. Abolfazl Hosseini and Parya Farahmand. "An attack resistant hybrid blind image watermarking scheme based on combination of DWT, DCT and PCA". In: *Multim. Tools Appl.* 83.7 (2024), pp. 18829–18852. DOI: 10.1007/S11042-023-16202-2.

[HKB09] Amir Houmansadr, Negar Kiyavash, and Nikita Borisov. "RAINBOW: A Robust And Invisible Non-Blind Watermark for Network Flows". In: *Proceedings of the Network and Distributed System Security Symposium, NDSS*. The Internet Society, 2009. URL: https://www.ndss-symposium.org/ndss2009/rainbow-a-robust-and-invisible-non-blind-watermark-for-network-flows/.

[HPC10] Raju Halder, Shantanu Pal, and Agostino Cortesi. "Watermarking Techniques for Relational Databases: Survey, Classification and Comparison". In: *J. Univers. Comput. Sci.* (2010), pp. 3164–3190.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/NECO.1997.9.8.1735.

[Jia+23] Renhe Jiang, Zekun Cai, Zhaonan Wang, Chuang Yang, Zipei Fan, Quanjun Chen, Kota Tsubouchi, Xuan Song, and Ryosuke Shibasaki. "DeepCrowd: A Deep Model for Large-Scale Citywide Crowd Density and Flow Prediction". In: *IEEE Trans. Knowl. Data Eng.* 35.1 (2023), pp. 276–290. DOI: 10.1109/TKDE.2021.3077056.

[Jin+05] Xiaoming Jin, Zhihao Zhang, Jianmin Wang, and Deyi Li. "Watermarking Spatial Trajectory Database". In: *Database Systems for Advanced Applications, 10th International Conference, DASFAA*. Vol. 3453. Lecture Notes in Computer Science. Springer, 2005, pp. 56–67. DOI: 10.1007/11408079_8.

[JZH21] Christian Janiesch, Patrick Zschech, and Kai Heinrich. "Machine learning and deep learning". In: *Electronic Markets* 31.3 (2021), pp. 685–695. DOI: 10.1007/s12525-021-00475-2.

[KJ16] Hari Krishna Kanagala and V.V. Jaya Rama Krishnaiah. "A comparative study of K-Means, DBSCAN and OPTICS". In: *2016 International Conference on Computer Communication and Informatics (ICCCI)*. 2016, pp. 1–6. DOI: 10.1109/ICCCI.2016.7479923.

[Koh95] Teuvo Kohonen. *Self-Organizing Maps*. Vol. 30. Springer Series in Information Sciences. Springer, 1995. ISBN: 978-3-642-97612-4. DOI: 10.1007/978-3-642-97610-0.

[KSD11] Vivekananda Bhat K., Indranil Sengupta, and Abhijit Das. "A New Audio Watermarking Scheme Based on Singular Value Decomposition and Quantization". In: *Circuits Syst. Signal Process.* 30.5 (2011), pp. 915–927. DOI: 10.1007/S00034-010-9255-8.

[KW17] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *5th International Conference on Learning Representations, ICLR*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=SJU4ayYgl.

[LeC+98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition". In: *Proc. IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[LFK09] Andrea Lancichinetti, Santo Fortunato, and János Kertész. "Detecting the overlapping and hierarchical community structure in complex networks". In: *New journal of physics* 11.3 (2009), p. 033015. DOI: 10.1088/1367-2630/11/3/033015.

[Li+22a] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects". In: *IEEE Trans. Neural Networks Learn. Syst.* 33.12 (2022), pp. 6999–7019. DOI: 10.1109/TNNLS.2021.3084827.

[Li+22b] Zhonghang Li, Chao Huang, Lianghao Xia, Yong Xu, and Jian Pei. "Spatial-Temporal Hypergraph Self-Supervised Learning for Crime Prediction". In: *38th IEEE International Conference on Data Engineering, ICDE*. IEEE, 2022, pp. 2984–2996. DOI: 10.1109/ICDE53745.2022.00269.

[Li+23] Yi Li, Weiming Huang, Gao Cong, Hao Wang, and Zheng Wang. "Urban Region Representation Learning with OpenStreetMap Building Footprints". In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD*. ACM, 2023, pp. 1363–1373. DOI: 10.1145/3580305.3599538.

[Li+24] Zechen Li, Weiming Huang, Kai Zhao, Min Yang, Yongshun Gong, and Meng Chen. "Urban Region Embedding via Multi-View Contrastive Prediction". In: *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI*. AAAI Press, 2024, pp. 8724–8732. DOI: 10.1609/AAAI.V38I8.28718.

[Liu+12] Yu Liu, Chaogui Kang, Song Gao, Yu Xiao, and Yuan Tian. "Understanding intra-urban trip patterns from taxi trajectory data". In: *J. Geogr. Syst.* 14.4 (2012), pp. 463–483. DOI: 10.1007/S10109-012-0166-Z.

[Luo+23] Xiyang Luo, Yinxiao Li, Huiwen Chang, Ce Liu, Peyman Milanfar, and Feng Yang. "DVMark: A Deep Multiscale Framework for Video Watermarking". In: *IEEE Transactions on Image Processing* (2023), pp. 1–1. DOI: 10.1109/TIP.2023.3251737.

[Mac67] J Macqueen. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*. 1967.

[Mon+23] Amin Karimi Monsefi, Pouya Shiri, Ahmad Mohammadshirazi, Nastaran Karimi Monsefi, Ron Davies, Sobhan Moosavi, and Rajiv Ramnath. "CrashFormer: A Multimodal Architecture to Predict the Risk of Crash". In: *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Advances in Urban-AI, UrbanAI 2023, Hamburg, Germany, 13 November 2023*. ACM, 2023, pp. 42–51. DOI: 10.1145/3615900.3628769.

[Moo+19] Sobhan Moosavi, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, Radu Teodorescu, and Rajiv Ramnath. "Accident Risk Prediction based on Heterogeneous Sparse Data: New Dataset and Insights". In: *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL*. ACM, 2019, pp. 33–42. DOI: 10.1145/3347146.3359078.

[NMP24] Kamran I. Naqash, Shahid A. Malik, and Shabir A. Parah. "Robust Audio Watermarking Based on Iterative Filtering". In: *Circuits Syst. Signal Process.* 43.1 (2024), pp. 348–367. DOI: 10.1007/S00034-023-02475-3.

[Nom] Nominatim. https://nominatim.org/.

[Nus81] Henri J Nussbaumer. "The fast Fourier transform". In: *Fast Fourier Transform and Convolution Algorithms*. Springer, 1981, pp. 80–111.

[OSMa] OSM. https://www.openstreetmap.org/.

[OSMb] Planet OSM. https://wiki.openstreetmap.org/wiki/Planet.osm.

[Pan+19] Zheyi Pan, Jie Bao, Weinan Zhang, Yong Yu, and Yu Zheng. "TrajGuard: A Comprehensive Trajectory Copyright Protection Scheme". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD*. ACM, 2019, pp. 3060–3070. DOI: 10.1145/3292500.3330685.

[PMB13] Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks". In: *Proceedings of the 30th International Conference on Machine Learning, ICML*. Vol. 28. JMLR Workshop and Conference Proceedings. JMLR.org, 2013, pp. 1310–1318. URL: http://proceedings.mlr.press/v28/pascanu13.html.

[RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning internal representations by error propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. MIT Press, 1986, pp. 318–362. ISBN: 026268053X. DOI: 10.1007/s11042-023-15608-2.

[Sar21] Iqbal H. Sarker. "Machine Learning: Algorithms, Real-World Applications and Research Directions". In: *SN Comput. Sci.* 2.3 (2021), p. 160. DOI: 10.1007/S42979-021-00592-X.

[SBG23] Mehri Salayani, Behzad Bakhtiari, and Seyed Hossein Ghafarian. "A Robust Zero-Watermarking for Audio Signal Using Supervised Learning". In: *Circuits Syst. Signal Process.* 42.6 (2023), pp. 3668–3705. DOI: 10.1007/S00034-022-02288-W.

[Sca+09] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. "The Graph Neural Network Model". In: *IEEE Trans. Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[SHD23] Stefan Schestakov, Paul Heinemeyer, and Elena Demidova. "Road Network Representation Learning with Vehicle Trajectories". In: *Advances in Knowledge Discovery and Data Mining - 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD*. Vol. 13938. Lecture Notes in Computer Science. Springer, 2023, pp. 57–69. DOI: 10.1007/978-3-031-33383-5_5.

[She+23] Jingyi Shen, Haoyu Li, Jiayi Xu, Ayan Biswas, and Han-Wei Shen. "IDLat: An Importance-Driven Latent Generation Method for Scientific Data". In: *IEEE Trans. Vis. Comput. Graph.* 29.1 (2023), pp. 679–689. DOI: 10.1109/TVCG.2022.3209419.

[SS20] Dalwinder Singh and Birmohan Singh. "Investigating the impact of data normalization on classification performance". In: *Appl. Soft Comput.* 97.Part B (2020), p. 105524. DOI: 10.1016/J.ASOC.2019.105524.

[Su+20] Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. "A survey of trajectory distance measures and performance evaluation". In: *VLDB J.* (2020), pp. 3–32.

[Sun+24] Tianao Sun, Ke Fu, Weiming Huang, Kai Zhao, Yongshun Gong, and Meng Chen. "Going Where, by Whom, and at What Time: Next Location Prediction Considering User Preference and Temporal Regularity". In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD*. ACM, 2024, pp. 2784–2793. DOI: 10.1145/3637528.3671916.

[TL19] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *Proceedings of the 36th International Conference on Machine Learning, ICML*. Vol. 97. PMLR, 2019, pp. 6105–6114. URL: http://proceedings.mlr.press/v97/tan19a.html.

[Var+19] John E. Vargas-Muñoz, Sylvain Lobry, Alexandre X. Falcão, and Devis Tuia. "Correcting rural building annotations in OpenStreetMap using convolutional neural networks". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 147 (2019), pp. 283–293. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2018.11.010.

[Vel+18] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. "Graph Attention Networks". In: *6th International Conference on Learning Representations, ICLR*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=rJXMpikCZ.

[Ver+21] Muskan Verma, Nayan Gupta, Bhavishya Tolani, and Rishabh Kaushal. "Explainable Custom CNN Architecture for Land Use Classification using Satellite Images". In: *2021 Sixth International Conference on Image Information Processing (ICIIP)*. Vol. 6. 2021, pp. 304–309. DOI: 10.1109/ICIIP53038.2021.9702698.

[VR04] AK Verma and S Rajotia. "Feature vector: a graph-based feature recognition methodology". In: *International journal of production research* 42.16 (2004), pp. 3219–3234. DOI: 10.1080/00207540410001699408.

[Win78] Shmuel Winograd. "On computing the discrete Fourier transform". In: *Mathematics of computation* (1978), pp. 175–199.

[Wow+22] Kelvin Sopnan Wowo, Rajjat Dadwal, Timo Graen, Andrea Fiege, Michael Nolting, Wolfgang Nejdl, Elena Demidova, and Thorben Funke. "Using Vehicle Data to Enhance Prediction of Accident-Prone Areas". In: *25th IEEE International Conference on Intelligent Transportation Systems, ITSC*. IEEE, 2022, pp. 2450–2456. DOI: 10.1109/ITSC55140.2022.9922236.

[WS21] Szymon Wozniak and Piotr Szymanski. "hex2vec: Context-Aware Embedding H3 Hexagons with OpenStreetMap Tags". In: *GeoAI@SIGSPATIAL 2021: Proceedings of the 4th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*. ACM, 2021, pp. 61–71. DOI: 10.1145/3486635.3491076.

[Wu+22] Shangbin Wu, Xu Yan, Xiaoliang Fan, Shirui Pan, Shichao Zhu, Chuanpan Zheng, Ming Cheng, and Cheng Wang. "Multi-Graph Fusion Networks for Urban Region Embedding". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*. ijcai.org, 2022, pp. 2312–2318. DOI: 10.24963/IJCAI.2022/321.

[Xi+22] Yanxin Xi, Tong Li, Huandong Wang, Yong Li, Sasu Tarkoma, and Pan Hui. "Beyond the First Law of Geography: Learning Representations of Satellite Imagery by Leveraging Point-of-Interests". In: *WWW '22: The ACM Web Conference*. ACM, 2022, pp. 3308–3316. DOI: 10.1145/3485447.3512149.

[XL23] Hui Xie and PengYu Liang. "Improved TF-LSTM Multi-Step Vehicle Speed Prediction Model Based on LSTM and Attention Mechanism". In: *2023 3rd International Conference on Robotics, Automation and Intelligent Control (ICRAIC)*. 2023, pp. 436–440. DOI: 10.1109/ICRAIC61978.2023.00082.

[Yam+18] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology". In: *Insights into imaging* 9 (2018), pp. 611–629. DOI: 10.1007/s13244-018-0639-9.

[Yan+13] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhiwen Yu. "Fine-grained preference-aware location search leveraging crowdsourced digital footprints from LBSNs". In: *The 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '13, 2013*. ACM, 2013, pp. 479–488. DOI: 10.1145/2493432.2493464.

[YZY18] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. "Hetero-ConvLSTM: A Deep Learning Approach to Traffic Accident Prediction on Heterogeneous Spatio-Temporal Data". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD*. ACM, 2018, pp. 984–992. DOI: 10.1145/3219819.3219922.

[Zha+20] Mingyang Zhang, Tong Li, Yong Li, and Pan Hui. "Multi-View Joint Graph Representation Learning for Urban Region Embedding". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*. ijcai.org, 2020, pp. 4431–4437. DOI: 10.24963/IJCAI.2020/611.

[Zha22] Liang Zhao. "Event Prediction in the Big Data Era: A Systematic Survey". In: *ACM Comput. Surv.* 54.5 (2022), 94:1–94:37. DOI: 10.1145/3450287.

[Zho+20] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. "Graph neural networks: A review of methods and applications". In: *AI Open* 1 (2020), pp. 57–81. DOI: 10.1016/J.AIOPEN.2021.01.001.

[Zho+23] Silin Zhou, Dan He, Lisi Chen, Shuo Shang, and Peng Han. "Heterogeneous Region Embedding with Prompt Learning". In: *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI*. AAAI Press, 2023, pp. 4981–4989. DOI: 10.1609/AAAI.V37I4.25625.

[ZLC23] Liang Zhang, Cheng Long, and Gao Cong. "Region Embedding With Intra and Inter-View Contrastive Learning". In: *IEEE Trans. Knowl. Data Eng.* 35.9 (2023), pp. 9031–9036. DOI: 10.1109/TKDE.2022.3220874.

Appendices

Appendix A

Publication: An Adaptive Clustering Approach for Accident Prediction

Rajjat Dadwal, Thorben Funke, and Elena Demidova

Intelligent Transportation Systems Conference (ITSC), 2021

DOI: 10.1109/ITSC48978.2021.9564564

An Adaptive Clustering Approach for Accident Prediction

Rajjat Dadwal[1], Thorben Funke[1], Elena Demidova[2]

Abstract— **Traffic accident prediction is a crucial task in the mobility domain. State-of-the-art accident prediction approaches are based on static and uniform grid-based geospatial aggregations, limiting their capability for fine-grained predictions. This property becomes particularly problematic in more complex regions such as city centers. In such regions, a grid cell can contain subregions with different properties; furthermore, an actual accident-prone region can be split across grid cells arbitrarily. This paper proposes Adaptive Clustering Accident Prediction (ACAP) - a novel accident prediction method based on a grid growing algorithm. ACAP applies adaptive clustering to the observed geospatial accident distribution and performs embeddings of temporal, accident-related, and regional features to increase prediction accuracy. We demonstrate the effectiveness of the proposed ACAP method using open real-world accident datasets from three cities in Germany. We demonstrate that ACAP improves the accident prediction performance for complex regions by 2-3 percent points in F1-score by adapting the geospatial aggregation to the distribution of the underlying spatio-temporal events. Our grid growing approach outperforms the clustering-based baselines by four percent points in terms of F1-score on average.**

I. Introduction

Prediction of traffic accidents is an important research area in the mobility, urban safety, and city planning domains. Such prediction is particularly challenging due to the data sparsity, the complexity of the spatio-temporal event distribution, the variety of the involved influence factors, and the complexity of their relationships.

State-of-the-art accident prediction methods (e.g., [1], [2]) mainly focus on two prediction aspects, namely feature selection to identify relevant influence factors and the definition of the predictive model architecture. One crucial aspect, typically neglected by the existing works, is the geospatial aggregation underlying predictive models. Whereas some urban areas, such as city centers, have a more complex structure and tend to attract more accidents, other areas are less accident-prone. Hence, differently from existing works, we include geospatial aggregation as an essential factor in our modeling. Overall, we consider the spatio-temporal accident prediction problem according to the three dimensions: geospatial aggregation, feature selection, and predictive model architecture.

The forecasting of spatio-temporal accidents is particularly challenging due to data sparsity. Existing works address the data sparsity by adopting coarse geospatial aggregations, such as fixed grids [3] or entire administrative districts [4], as prediction targets. However, neither predefined grid cells nor administrative districts adequately fit the spatio-temporal distribution of the observed events. Furthermore, existing works on traffic accident prediction usually consider accident datasets in US cities (e.g., [1], [2]). These cities exhibit a grid-like structure, whereas European cities have the least grid-like structure [5], such that the models developed for the US cities are not directly applicable to Europe.

In this paper, we propose Adaptive Clustering Accident Prediction (*ACAP*) – a novel approach to infer adaptive grids from the observed sparse spatio-temporal event distributions. We perform predictions on adaptive task-specific regions obtained through the proposed clustering-based grid growing method. As a predictive model, we rely on a neural network approach. We combine time series forecasting, in the form of Gated Recurrent Units (GRUs), with an embedding of static regional features. Through experiments on real-world datasets, we demonstrate that the proposed method increases the prediction accuracy compared to the state-of-the-art baselines based on fixed grids. As our experiments demonstrate, our Adaptive Clustering Accident Prediction approach outperforms several machine learning and neural network baselines regarding F1-score on the accident prediction task in several cities in Germany.

We observed that most existing works focused on evaluating the model performance based on private datasets (e.g., [3]), which makes them difficult to reproduce and to extend by other researchers. We aim to foster reproducibility, reuse, and extensibility of our work by the research community. Hence, we use only publicly available open datasets as a basis for feature extraction. For example, we collect the regional attributes, such as street types or the number of junctions in a region, from OpenStreetMap (OSM)[1] - the largest publicly available source of map data. Furthermore, we build our accident prediction model on the "German Accident Atlas"[2] – a publicly available official dataset containing traffic accident data for Germany. Moreover, we make our data processing pipeline available open-source[3].

In summary, our contributions are as follows:

1) We propose *ACAP* – a novel approach to infer adaptive grids from sparse spatio-temporal accident distributions.
2) Our proposed prediction model using *ACAP* as geospatial aggregation achieves state-of-the-art prediction performance on the general task of traffic accident predic-

[1]L3S Research Center, Leibniz University Hannover, Appelstraße 9a, 30167 Hannover, Germany dadwal@L3S.de, tfunke@L3S.de

[2]Data Science & Intelligent Systems (DSIS) Research Group, University of Bonn, Friedrich-Hirzebruch-Allee 5, 53115 Bonn, Germany demidova@cs.uni-bonn.de

[1]OpenStreetMap: https://www.openstreetmap.org/

[2]Accident data: https://unfallatlas.statistikportal.de

[3]Software: https://github.com/Rajjat/ACAP

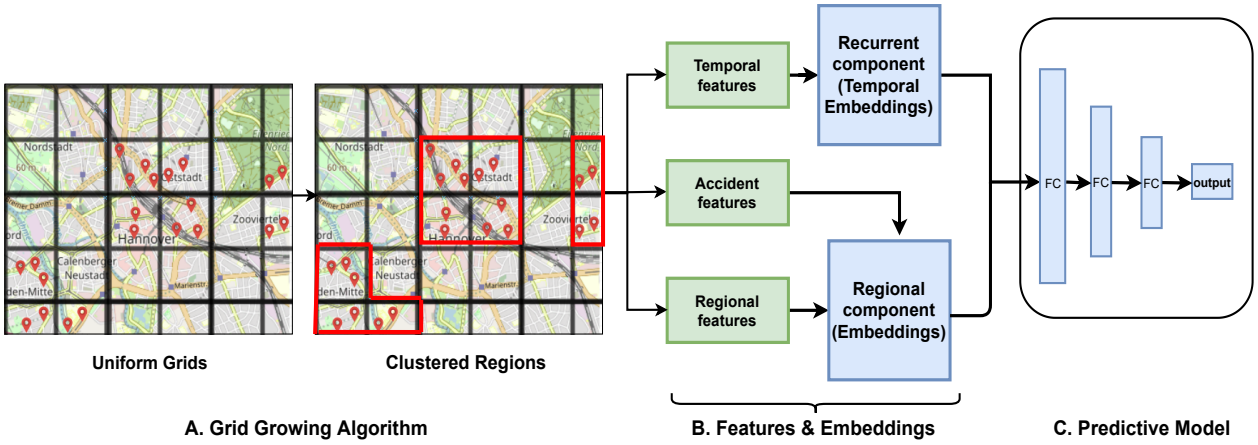


Fig. 1. Architecture of the proposed *ACAP* approach: a) Adaptive Clustering b) Features + Embeddings c) Classification/Prediction

tion and significantly improves the prediction results in the more complex areas such as city centers.

3) Our *ACAP* approach relies on open data and an open-source pipeline.
4) Our experiments demonstrate that *ACAP* outperforms several baselines with a performance increase of 2-3% on average concerning F1-score on three large German cities.

The rest of the paper is structured as follows: First, we discuss related work in Section II. Then, in Section III we present the formal problem statement for sparse spatio-temporal event prediction. In Section IV, we present our proposed *ACAP* approach based on adaptive clustering with grid growing. Section V describes our experimental setup, including baselines and datasets. We present the evaluation results on open real-world datasets in Section VI. Finally, we provide a conclusion in Section VII.

II. Related Work

In this section, we discuss related work on accident prediction. While existing approaches perform accident prediction on fixed grids or specific highways/streets, the proposed *ACAP* approach adapts to the specific regions. In the following, we discuss relevant accident prediction approaches according to the spatial aggregations they adopt.

Prediction on fixed map grids. Moosavi *et al.* [2] developed a DAP model for predicting the occurrences of an accident on the 5x5 grid in a 15-minute interval. They evaluated the model on sparse data by augmenting it with the Point of Interests (POIs), weather, and time. Hetero-ConvoLSTM [1] predicted the number of accidents on the 5x5 grid during each time slot (a day). They used heterogeneous data, including roads, weather, time, traffic, and satellite images. Ren *et al.* [6] employed an LSTM model that predicts the frequency of accidents, given the history of the past 100 hours, for 1x1 grids. In another study by Chen *et al.* [3], the accident prediction is performed on a $500\text{m}\times500\text{m}$ grid cell with the human mobility data as well as a set of 300,000 accident records in Tokyo (Japan). The authors predicted the possibility of accident occurrence on an hourly basis.

Prediction on street segments. The works in this category deal with predicting an accident or accident count on a given road/highway. Chang *et al.* [7] used information such as road geometry, annual average daily traffic and weather data to predict the frequency of accidents for a highway in Taiwan using a neural network and compared the results with the Poisson or negative binomial regression. Caliendo *et al.* [8] embedded road attributes such as length, curvature, annual average daily traffic, sight distance, side friction coefficient, longitudinal slope, and the presence of a junction to predict the accident count on a four-lane median-divided Italian freeway. There are similar works related to accident prediction on highways. For example, an accident prediction model by Wenqi *et al.* [9] based on a convolution neural network is designed to forecast an accident on the I-15 USA highway. Yuan *et al.* [10] predicted the accident occurrence for each road segment in the state of Iowa each hour in similar work. Hollenstein *et al.* [11] investigated the association of bicycle accident occurrence to roundabout properties of the road at Swiss roundabouts using a logistic regression approach. The authors also studied various features of roundabouts responsible for bicycle accidents.

In summary, existing approaches rely on a fixed grid of arbitrary size or a pre-defined street-segment aggregation. In contrast, *ACAP* is a novel adaptive approach for predicting accidents in spatially closed regions, irrespective of the fixed grids or specific street segments. Furthermore, *ACAP* works on sparse data, publicly available and easy to collect, in contrast to the approaches that use extensive but often closed datasets for modeling and prediction.

III. Problem Statement

We phrase our considered problem of traffic accident prediction in a general fashion of sparse spatio-temporal event prediction. Since in this paper we are only interested in predicting traffic accidents, as a particular case of spatio-temporal events, we use events and accidents as synonyms.

Let $\mathbf{E} \subset \mathbb{R}^3$ be the set of spatio-temporal events, i.e., each event $E \in \mathbf{E}$ consists of the latitude, longitude, and time information. We are interested in the prediction of these events for different spatial aggregations: Let $f\colon \mathbb{R}^3 \to \mathbb{N}^2$ be the aggregation function mapped into $R_{\max}$ cells and $T_{\max}$ time intervals, i.e., $f(\mathbf{E}) \subset \{0, \ldots, R_{\max}\} \times \{0, \ldots, T_{\max}\}$. We are especially interested in studying the effect of different geospatial aggregations on prediction performance.

Since time and position do not provide sufficient information for developing predictive models in this domain, we assume additional features about each spatial cell and time interval. Formally, let $X_{\text{temporal}} \subset \mathbb{R}^{R_{\max} \times T_{\max} \times d_t}$ and $X_{\text{cells}} \subset \mathbb{R}^{R_{\max} \times d_r}$ be the matrices of the d_t temporal and d_r spatial features. For example, we have as part of the regional information X_{cells} the number of junctions, the street length, and the region size. Examples of region-specific temporal features X_{temporal} are solar elevation and solar azimuth.

Our task is to create a binary forecast based on k-historic observations, i.e., to train a function $\Phi\colon \{0, \ldots, R_{\max}\} \times \mathbb{R}^{d_r} \times \mathbb{R}^{k \times d_t} \to \{0, 1\}$ such that Φ outputs 1, if an event is observed in the next time period in the specific region, and 0 otherwise. We assume an imbalanced event set, where the occurrence of one event, e.g., non-accident, is much more likely than the other kind of event, e.g., accident. Furthermore, we are interested in comparing the performance over different spatial aggregations. Hence, it leads to change of the aggregation function $f\colon \mathbb{R}^3 \to \mathbb{N}^2$ to another aggregation function $\tilde{f}\colon \mathbb{R}^3 \to \mathbb{N}^2$.

IV. Approach

This section presents the Adaptive Clustering Accident Prediction (*ACAP*) approach proposed in this paper. The model architecture of *ACAP* is illustrated in Fig. 1. First, we propose an adaptive clustering technique to build clusters that reflect the geospatial distribution of the accidents, presented in Section IV-A. Then, our method generates temporal and geospatial feature embeddings, presented in Section IV-B. Finally, we describe the predictive model of *ACAP* in Section IV-C.

A. Adaptive Clustering with Grid Growing

Existing accident prediction approaches apply either a uniform geospatial aggregation using standard methods, such as geohash [12], or utilize administrative districts as a prediction target. The geospatial aggregation adopted by these approaches is often enforced by the already aggregated raw data, e.g., resulting from anonymization. The resulting uniform spatial grids are relatively coarse and do not reflect the actual accident distribution. Furthermore, existing works typically utilize US datasets such as Large-Scale Traffic and Weather Events Dataset (LSTW)[4], and IOWADOT data[5] for the evaluation. In these datasets, the uniform grid structure appears meaningful, as it follows the typical layout of the US cities. In contrast, the European cities' road layout does not typically follow the grid-like structure [5]. These observations motivate us to perform adaptive clustering to create geospatial aggregations that better fit the road layout and city infrastructure in the target region.

Algorithm 1 presents an overview of the adaptive clustering approach proposed in this work. This algorithm is based on our variant of grid growing [13], which learns geospatial regions based on the training data, e.g., past observed accidents. The algorithm includes two main steps: 1) grid construction and 2) grid growing. The grid construction step requires an initial geospatial grid as a basis. This grid is then aggregated iteratively to form larger regions that follow the event distribution. In the grid growing approach proposed by [13], the initial number of rows and columns is user-defined, and these parameters are not intuitive. In contrast, we construct the grid in a novel way with the help of geohash. Geohash encodes a geographic location into a string of letters and digits. Each character in the geohash defines a specific grid, e.g., "u1qcvmz82kw" stands for Hannover city center. Longer geohash values correspond to the fine-granular grids with smaller cell sizes. In this work, we experiment with the geohash of length five, six, and seven, which approximately correspond to the regions of $4.89\text{km} \times 4.89\text{km}$ (5x5), $1.22\text{km} \times 0.61\text{km}$ (1x1), and $153\text{m} \times 153\text{m}$ (0.1x0.1), respectively. We experimentally assess the influence of the geohash length and utilize the geohash of length seven, which corresponds to the smaller cell size, i.e., 0.1x0.1 (δ_{detail}), in our grid growing approach.

The next step is the grid growing. In the first step, we randomly select a seed, i.e., a grid cell containing an accident. The region starts growing from the current seed by searching for accidents in the neighbor cells. As the eight-neighbors search gives more accurate results than the four-neighbors search [13], we perform an eight-neighbors search to obtain nearby accidents in all adjacent grid cells. The grid growing stops when the current region does not find any accidents in the adjacent grid cells and assigns a cluster to the resulting region. In the next step, we choose the next seed cell randomly from the accident-prone grid cells not clustered in the previous algorithm iterations. The grid growing algorithm continues until it assigns all accident-prone grid cells in the training set to a cluster. Based on the clusters generated by the grid growing algorithm, we can, later on, assign locations and accidents unseen during training to their nearest clusters. To define the nearest cluster, we adopt haversine distance and apply a distance threshold Δ. We experimentally set $\Delta = 400$ meters. For the accident locations not mapped to any of the clusters due to the distance value exceeding the threshold, we map those locations to a larger base grid cell of 1x1 (δ_{base}) and assign this cell to a separate geospatial cluster.

The grid growing algorithm illustrated in Fig. 1 is essential for building adaptive regions. We compare the proposed grid growing approach to fixed grids and clustering approaches in the evaluation. The advantages of adaptive clustering, and especially of the grid growing approach proposed in this work, are as follows: (i) Our geospatial aggregation adapts to the underlying distribution of accidents in the dataset. In

[4] https://smoosavi.org/datasets/lstw

[5] https://public-iowadot.opendata.arcgis.com/datasets/crash-data

Algorithm 1 Adaptive Clustering with Grid Growing

Input: Spatio-(temporal) events $\mathbf{E}$, e.g., training set of accidents
Output: Spatial-aggregation function f_{GG}
Hyperparameters: detailed grid size δ_{detail}, base grid δ_{base}, distance threshold Δ
Calculate for each $E \in \mathbf{E}$ their detailed grid $G^{\delta_{\text{detail}}}(E)$
Initialize clusterings $\mathbf{C} = \emptyset$ and $i = 0$
while Unmarked event $E \in \mathbf{E}$ exist **do**
    Select random unmarked event $E \in \mathbf{E}$
    Set $C_i = \{E\}$
    repeat
        Check for each event in C_i the 8-$G^{\delta_{\text{detail}}}$-neighborhood for events $\mathbf{E}_{\text{neighbors}}$
        Set $C_i = C_i \cup \mathbf{E}_{\text{neighbors}}$
    until No new neighbors, i.e., $\mathbf{E}_{\text{neighbors}} = \emptyset$
    Mark all events in C_i and set $\mathbf{C} = \{C_0, \dots C_i\}$
end while
return $f_{\text{GG}}(E) = \begin{cases} C, & \text{if } C = \operatorname{argmin}_{\tilde{C} \in \mathbf{C}} \mathrm{d}(E, \tilde{C}), \\ & \text{and } \mathrm{d}(E, C) < \Delta, \\ G^{\delta_{\text{base}}}(E) & \text{otherwise} \end{cases}$

other words, we adjust the geospatial resolution based on the events that occur in the geospatial proximity. (ii) Our adaptive clustering allows us to work with sparse spatio-temporal data, unlike other baselines [2]. This property makes our approach easily applicable to large (rural) areas where the data can be extremely sparse.

B. Features & Embeddings

As a data pre-processing step, we compute temporal and geospatial features such as accident and regional features for each adaptive cluster and each grid cell. We evaluate the adaptive clustering approach with a fixed grid of cell size 5x5 and 1x1 in Section VI.

Temporal Features. Accidents are time-dependent, such that we aim to learn the correlation between the accidents and the temporal features. Our model includes ten temporal features such as weekday/weekend, season, month, year, weekdays, an hour of the day, daylight, solar position, solar azimuth, and solar elevation. All temporal features are encoded in one feature vector using the one-hot-encoding technique. The resulting feature vector includes 36 dimensions, where each dimension represents a possible feature value. The degree of temporal aggregation depends, in general, on data availability. In the "German Accident Atlas" dataset used in the evaluation, temporal features are aggregated on an hourly basis due to legal restrictions.

Accident Features. The accident features include the accident type and the road conditions during the accident. Examples of accident types in the "German Accident Atlas" dataset include a car collision with another car or a bicycle. A specific accident type can be more prominent at one location than others, e.g., a city center has more car collisions than collisions with a bicycle. Thus, the accident type feature helps to identify such areas. Road conditions feature informs whether the road was wet, slippery, or dry during an accident. The accident features are converted into one-hot-encoded vectors and averaged for the accidents in a geospatial cluster or a grid cell.

Regional Features. Regional features are infrastructural attributes of a specific region, i.e., a grid cell or an adaptive cluster. Intuitively, regional features have a significant influence on accident occurrences. For example, accidents tend to occur more often near junctions or crossings. We select the following Point of Interests (POIs) as regional features: amenities count, number of crossings, number of junctions, number of railways, station frequency, stop signs count, number of traffic signals, number of turning loops, number of giveaways, highway types, and the average maximum speed for each region. We normalize feature values to the range between 0 and 1. We extract regional features from OSM.

Feature Embedding. Embeddings are continuous vector representations of discrete variables. Embeddings can help to reduce the dimensionality of feature vectors and to represent latent features. We construct latent representations from one-hot-encoded and normalized feature vectors generated above as follows.

Temporal embeddings. For the temporal features, we utilize Gated Recurrent Unit (GRU) to create temporal embeddings. GRU is a type of Recurrent Neural Network (RNN) to learn sequential or temporal data.

A set of eight temporally ordered one-hot-encoded vectors from the preceding time points, each of length n, where n corresponds to the number of one-hot-encoded features, are fed to the GRU. With the temporal features listed above, n=36. GRU includes two recurrent layers in our settings, each with 128 units, and outputs the embedding vector of the same length.

Embeddings of accident and regional features. For these features, a feed-forward layer of size 128 with the sigmoid activation function creates feature embeddings.

C. Predictive Model

The predictive model of *ACAP* outputs a softmax, i.e., the likelihood for accidents respectively non-accidents. We transform them into binary accident labels, i.e., '1' for accident and '0' for non-accident. The model input is composed of the temporal embeddings and the embeddings of the accident and regional features of each geospatial cluster or grid cell. The input is feed-forwarded through the neural network layers with decreasing dimensionality. In particular, we use a set of fully connected layers of size 512, 256, 64, and 2, respectively. The activation function is applied in each layer to induce non-linearity in the model. The first three layers utilize ReLU as the activation function, whereas we apply softmax activation to the last layer's output. We use batch normalization [14] after the second and third layers. The role of batch normalization is to re-scale and normalize the intermediate outputs. The last layer is the classification layer that predicts binary accident labels. We optimize *ACAP* using categorical cross-entropy as a loss function.

V. Evaluation Setup

In this section we describe the baselines, datasets, parameters and metrics utilized in the evaluation.

A. Accident Prediction Baselines

We utilize four baseline methods, including machine learning and deep learning baselines: *Logistic Regression (LR), Gradient Boosting Classifier (GBC), Deep Neural Network (DNN), and Deep Accident Prediction (DAP)* model [2] to compare the performance of our approach regarding accident prediction.

LR is widely used for classification tasks where the model outputs probabilities for classification problems. GBC, another ML-based baseline with boosting characteristics, is also suitable for our classification task.

To compare our approach with deep learning models, we use DAP and DNN. DAP utilizes Long Short-Term Memory (LSTM) for temporal learning, Glove2Vec for learning accident descriptions, and embedding components for learning spatial attributes. DNN employs a set of fully connected layers of size 512, 256, 64, and 2, respectively.

B. Clustering Baselines

Geospatial aggregation can be broadly divided into two parts: grid-based and clustering-based. For the grid-based aggregation, we use the 5x5 and more detailed 1x1 geohash grids, as described in Section IV-A. The clustering approaches belong to the three categories: neural network-based, density-based, and centroid-based. As a representative of the neural network-based clustering methods, we evaluated Self-Organizing Map (SOM) [15], [16], [17]. In density-based clustering, DBSCAN [18] and its extension Hierarchical DBSCAN (HDBSCAN) have been used to cluster the geospatial data [13]. DBSCAN is an unsupervised machine learning algorithm to classify unlabeled data. As a representative of the centroid-based methods, we apply the well-known K-means algorithm [19].

C. Dataset

The accident dataset is collected by "The Federal Statistical Office" department in Germany and is openly accessible. This dataset includes accident information for 16 German federal states starting from 2016 and currently contains data until 2019. The dataset contains 24 accident attributes, including accident id, latitude, longitude, day of the week, hour, month, year, accident type, and road condition. Due to Germany's legal restrictions, the data is aggregated temporally on an hourly basis, and the specific date of the accident is not reported in the dataset. We filtered the dataset to obtain cities with a long observation period and a sufficient number of accidents to facilitate model training and selected Hannover, Munich, and Nuremberg. For example, Hannover and Nuremberg have comparable accidents count with 7,433 and 6,121, respectively. In contrast, Munich accounts for the highest number of accidents, with 14,986 accidents in the considered period.

OpenStreetMap Dataset. OSM is a publicly available geospatial database. One can easily extract and store regional features such as POIs from OSM geofabrik[6]. For example, around 50 percent of the accidents happened at primary, secondary, tertiary, and trunk highways in Lower Saxony, Germany. The aim is to leverage our model with regional features to help in the prediction task. We fetch the regional features from the OSM dataset, e.g., number of amenities, number of junctions, number of traffic signals, and different highway types. We aggregate each regional feature to its 0.1x0.1 geohash and map it to the clusters and grids in our settings.

Negative Samples. Accident prediction is a binary classification task that requires generating elements of the non-accident class. Any spatio-temporal point where no accident has occurred can be considered as a non-accident. However, using all time points leads to the generation of too many non-accidents. To compare different spatio-temporal aggregations on the same dataset, we randomly select a 0.1x0.1 geohash grid and randomly generate a temporal and spatial point for the selected grid. Motivated by [10], we maintain a fixed accident to non-accident ratio, i.e., 1:3 across training and test data.

Training and Test Split. We split three years of data into training and test data: first 29 months, i.e., 80% of data for training, and last seven months, i.e., 20% for testing. For validation, we utilize the hold-out cross-validation method. In this method, a subset (10%) of the training data (split temporally) is reserved for validating the model performance. The early stopping technique based on the validation set is performed as a regularization step with patience as an argument. Patience represents the number of epochs before stopping once the loss starts to increase. We train each model separately for each city and perform testing on the same city.

D. Hyperparameters

In the following, we describe the hyperparameter settings of the models adopted in the evaluation.

Clustering Baselines. We initialize the hyperparameters of the clustering baseline as follows. DBSCAN takes epsilon (e) and the minimum number of points (n) as input parameters. The value of e is determined by the DMDBSCAN algorithm [20] using the nearest neighbor search. The selected e with a combination of different values of n help to determine silhouette scores [21]. The values of e and n with the highest silhouette score are chosen. HDBSCAN has minimum cluster size as the only parameter, which we set to four. We apply the elbow method to determine the number of clusters in K-means (K=4). For SOM, we choose a map size of 30×30, which gives a comparable number of clusters as the 1x1 grid.

Model Hyperparameters. We find the best parameter setting for the aforementioned ML-based baseline models by using grid-search. We follow the same setting as in [2] and refer to our available code for further details about the baselines' hyperparameters.

For *ACAP*, Adam optimizer with an initial learning rate of 0.01 is used to train the model. A dropout of 0.2 is used for regularization in the GRU layer. In early stopping, patience with 15 helps in regularization. For DNN, the parameter

[6] https://download.geofabrik.de/europe/

setting is the same as in the fully connected predictive model of the *ACAP*. All the neural network-based models are trained for 60 epochs.

E. Evaluation Metric

Due to uneven class distribution, we use F1-score as a metric for evaluating different models. F1-score is the harmonic mean of precision and recall. Since we are interested in predicting the accident class, we report the F1-score of the accident class for different models. We run each model ten times and report the average F1-score.

VI. Evaluation

The evaluation aims to assess the proposed accident prediction approach, analyze the effect of the proposed adaptive geospatial clustering and examine feature importance.

A. Effect of Geospatial Clustering

As the first step of the *ACAP* evaluation, we compare different spatial clustering methods by changing the clustering in *ACAP*. In other words, we change the adaptive clustering (AC) part of our approach and plug in other clustering methods. Fig. 2 shows that our grid growing approach, i.e., GG outperforms all baselines by at least four percent points. The best performing baselines are SOM and DBSCAN, while HDBSCAN and K-means result in the worst model performance. Overall, we can observe that the proposed geospatial clustering has a significant positive effect on the observed performance. To further analyze our model and the geospatial aggregation, we evaluate *ACAP* and the best clustering baseline SOM against two uniform grids on three different cities in the next section.

B. General Performance

To extensively study *ACAP* performance, we evaluated *ACAP* using four different spatial aggregations and four other prediction methods on three German cities. As Table I shows, our *ACAP* approach achieves the highest F1-score in the accident prediction for all spatial aggregations and all cities. With respect to spatial aggregation, our grid growing clustering and 1x1 grids achieve the best results, while especially 5x5 grids reduce the prediction quality. We observe that the aggregation of static features in large uniform grids negatively impacts the performance and only achieves a one percent point higher score in Hannover than K-means, while having 31 regions instead of four. Overall, *ACAP* increases F1-score by 2-3 percent points over the best performing baseline on average.

C. Performance in the City Centers

To further analyze the proposed grid growing algorithm in urban regions, we evaluate the performance of our approach starting from the city center of Hannover to the larger Hannover region. For simplicity, we select a different radius around the city center of Hannover and compare the performance of grid growing and 1x1 grids. As Fig. 3 illustrates, *ACAP* with grid growing outperforms the uniform grids in the inner city center by 2 percent points.

TABLE I
F1-score of accident predictions of different cities with different aggregations

Clustering	Method	Hannover	Munich	Nuremberg
Grid-Growing	*ACAP*	**0.58**	**0.56**	**0.60**
Grid-Growing	DAP	0.47	0.44	0.47
Grid-Growing	DNN	0.55	0.52	0.54
Grid-Growing	LR	0.56	0.49	0.53
Grid-Growing	GBC	0.52	0.52	0.56
SOM	*ACAP*	**0.56**	**0.55**	**0.57**
SOM	DAP	0.42	0.44	0.45
SOM	DNN	0.51	0.51	0.54
SOM	LR	0.53	0.46	0.53
SOM	GBC	0.52	0.51	0.54
1x1	*ACAP*	**0.59**	**0.57**	**0.60**
1x1	DAP	0.49	0.49	0.51
1x1	DNN	0.57	0.52	0.57
1x1	LR	0.57	0.52	0.57
1x1	GBC	0.52	0.51	0.55
5x5	*ACAP*	**0.52**	**0.51**	**0.53**
5x5	DAP	0.45	0.44	0.45
5x5	DNN	0.50	0.49	0.51
5x5	LR	0.49	0.40	0.48
5x5	GBC	0.16	0.26	0.18

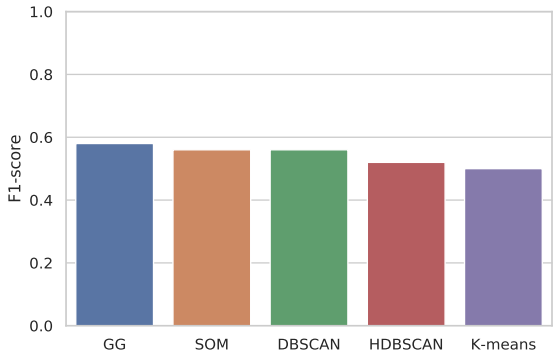


Fig. 2. Comparison of spatial clustering methods for Hannover city (*ACAP*)

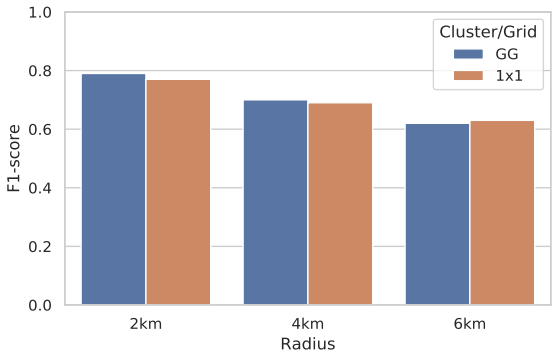


Fig. 3. F1-score vs radius from the city center of Hannover (*ACAP*)

D. Feature Importance

As the final part of our evaluation, we study the importance of our three feature groups – regional, temporal, and accident features – for *ACAP*'s performance. Fig. 4 shows the resulting accident F1-score if the model only uses one feature category for the prediction. We observe the high relevance of regional and temporal features, which achieve 91% and 63% of the model that relies on all features, correspondingly.

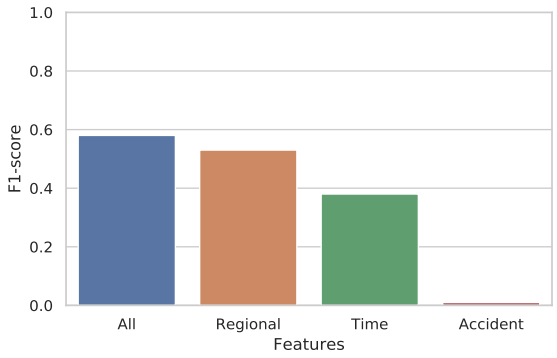


Fig. 4. Effect of different features (used alone) on F1-score for Hannover (*ACAP*)

VII. Conclusion

In this paper, we proposed *ACAP* – an approach that relies on novel adaptive clustering and various temporal and regional features to predict traffic accidents. Overall, we achieved a 2-3 percent points increase in F1-score over the best-performing baseline on average. Our proposed grid growing algorithm, which flexibly adapts to the regions based on the observed geospatial accident distribution, increases the performance by four percent points against the clustering-based baselines. We observed that our grid growing approach improves the prediction performance by two percent points in the city centers. Furthermore, *ACAP* is based on an open data pipeline, which comes with our publicly available implementation, making the proposed approach reproducible and reusable. In future work, we plan to investigate the impact of user-centric features, such as driver behavior, on accident prediction.

Acknowledgements

This work is partially funded by the BMWi, Germany under the projects "CampaNeo" (grant ID 01MD19007B), and "d-E-mand" (grant ID 01ME19009B), the European Commission (EU H2020, "smashHit", grant-ID 871477) and DFG, German Research Foundation ("WorldKG", DE 2299/2-1).

References

[1] Z. Yuan, X. Zhou, and T. Yang, "Hetero-ConvLSTM: A Deep Learning Approach to Traffic Accident Prediction on Heterogeneous Spatio-Temporal Data," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018*. ACM, 2018, pp. 984–992.

[2] S. Moosavi, M. H. Samavatian, S. Parthasarathy, R. Teodorescu, and R. Ramnath, "Accident risk prediction based on heterogeneous sparse data: New dataset and insights," in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2019*. ACM, 2019, pp. 33–42.

[3] Q. Chen, X. Song, H. Yamada, and R. Shibasaki, "Learning deep representation from big and heterogeneous data for traffic accident inference," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 2016, pp. 338–344.

[4] K. El-Basyouny and T. Sayed, "Accident prediction models with random corridor parameters," *Accident Analysis & Prevention*, vol. 41, no. 5, pp. 1118–1123, 2009.

[5] G. Boeing, "Urban spatial order: street network orientation, configuration, and entropy," *Appl. Netw. Sci.*, vol. 4, no. 1, pp. 67:1–67:19, 2019.

[6] H. Ren, Y. Song, J. Wang, Y. Hu, and J. Lei, "A deep learning approach to the citywide traffic accident risk prediction," in *Proceedings of the 21st International Conference on Intelligent Transportation Systems, ITSC 2018*. IEEE, 2018, pp. 3346–3351.

[7] L.-Y. Chang, "Analysis of freeway accident frequencies: negative binomial regression versus artificial neural network," *Safety science*, vol. 43, no. 8, pp. 541–557, 2005.

[8] C. Caliendo, M. Guida, and A. Parisi, "A crash-prediction model for multilane roads," *Accident Analysis & Prevention*, 2007.

[9] L. Wenqi, L. Dongyu, and Y. Menghua, "A model of traffic accident prediction based on convolutional neural network," in *Proceedings of the 2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. IEEE, 2017, pp. 198–202.

[10] Z. Yuan, X. Zhou, T. Yang, J. Tamerius, and R. Mantilla, "Predicting traffic accidents through heterogeneous urban data: A case study," in *Proceedings of the 6th International Workshop on Urban Computing (UrbComp 2017)*, vol. 14, 2017.

[11] D. Hollenstein, M. Hess, D. Jordan, and S. Bleisch, "Investigating roundabout properties and bicycle accident occurrence at swiss roundabouts: A logistic regression approach," *ISPRS Int. J. Geo Inf.*, vol. 8, no. 2, p. 95, 2019.

[12] G. M. Morton, "A computer oriented geodetic data base and a new technique in file sequencing," 1966.

[13] Q. Zhao, Y. Shi, Q. Liu, and P. Fränti, "A grid-growing clustering algorithm for geo-spatial data," *Pattern Recognit. Lett.*, vol. 53, pp. 77–84, 2015.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, vol. 37. JMLR.org, 2015, pp. 448–456.

[15] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proceedings of the IEEE*, vol. 84, no. 10, pp. 1358–1384, 1996.

[16] P. Mangiameli, S. K. Chen, and D. West, "A comparison of som neural network and hierarchical clustering methods," *European Journal of Operational Research*, vol. 93, no. 2, pp. 402–417, 1996.

[17] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 11, no. 3, pp. 586–600, 2000.

[18] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press, 1996, pp. 226–231.

[19] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[20] N. Rahmah and I. S. Sitanggang, "Determination of optimal epsilon (eps) value on DBSCAN algorithm to clustering data on peatland hotspots in sumatra," in *IOP Conference Series: Earth and Environmental Science*, vol. 31, no. 1. IOP Publishing, 2016, p. 012012.

[21] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

Appendix B

Publication: A Multimodal and Multitask Approach for Adaptive Geospatial Region Embeddings

Rajjat Dadwal, Ran Yu, and Elena Demidova

The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 2024

DOI: 10.1007/978-981-97-2262-4_29

A Multimodal and Multitask Approach for Adaptive Geospatial Region Embeddings

Rajjat Dadwal[1,2( )], Ran Yu[1,2], and Elena Demidova[1,2]

[1] Data Science and Intelligent Systems Group (DSIS), University of Bonn, Bonn, Germany
{dadwal,ran.yu,elena.demidova}@cs.uni-bonn.de
[2] Lamarr Institute for Machine Learning and Artificial Intelligence, Bonn, Germany
https://lamarr-institute.org

Abstract. Geospatial region embeddings are vital in developing predictive models tailored to urban environments. Such models enable critical applications, including crime rate prediction and land usage classification. However, state-of-the-art methods typically generate embeddings based on fixed administrative regions. These regions may not always align with specific tasks or areas of user interest. Creating fine-grained embeddings tailored to specific tasks and regions of user interest is labor-intensive and requires substantial resources. In this paper, we propose *MAGRE* – a novel approach that generates fine-granular adaptive geospatial region embeddings by leveraging multimodal and multitask learning. The embeddings generated by *MAGRE* can be flexibly aggregated to suit various region boundaries, rendering them effective in diverse urban applications. Our experimental results demonstrate that *MAGRE*'s embeddings outperform state-of-the-art embedding baselines, resulting in a 25.73% reduction in root mean squared error for crime rate prediction and a 19.08% reduction for check-in count prediction.

Keywords: Adaptive Geospatial Embeddings · Multitask Learning

1 Introduction

Real-world applications that rely on geographic data often require embeddings of the regions of interest (ROIs) for a particular user and a task. Geospatial region embeddings play an essential role in consolidating information across sources and enable capturing complex spatial relationships within and across regions. Such embeddings have proven beneficial in various applications, including land use classification and crime rate prediction [12,16]. However, embeddings created by existing methods may not align with the regions and tasks of user interest.

The mismatch between the embedding provided by the state-of-the-art methods and the ROIs results from the substantial limitations of the existing geospatial region embedding approaches. First, conventional geospatial embeddings rely on fixed administrative boundaries, such as districts [12,15]. Second,

D.-N. Yang et al. (Eds.): PAKDD 2024, LNAI 14649, pp. 363–375, 2024.
https://doi.org/10.1007/978-981-97-2262-4_29

Fig. 1. Manhattan division based on administrative boundaries (left) and based on hexagonal grids (right). The user ROIs are marked in orange. Map data: ©OpenStreetMap contributors, ODbL. (Color figure online)

geospatial embeddings created for smaller geometric-shape regions typically rely on the skip-gram model [11], which is unsuitable for embedding aggregation [1]. Third, existing approaches use satellite imagery [13] as multimodal contextual information. However, satellite imagery has limited accessibility and requires substantial data acquisition and preprocessing effort. Finally, region embeddings are often designed for specific tasks [12], neglecting significant factors of urban dynamics and patterns, and may fail to generalize to unseen tasks.

For example, Fig. 1 illustrates the Manhattan division based on administrative boundaries and smaller hexagonal-shaped grid cells. A user may be interested in assessing crime rates for property purchases for the ROIs (encoded in orange). The spatial misalignment between the user's ROIs and the precomputed region embeddings based on administrative boundaries may result in an inaccurate crime rate assessment. In contrast, the union of the grid-cell-based embeddings can capture the ROI more precisely.

This paper introduces a novel approach for obtaining geospatial region embeddings efficiently, focusing on adaptable regions of interest (ROIs). Our idea involves generating adaptive region embeddings by embedding smaller geospatial units (grid cells) and dynamically aggregating them into an ROI flexibly on demand. However, such aggregation is challenging. Due to limited data, the representation of individual grid cells might lack context and broad applicability. Furthermore, the semantics of the ROI as a whole may differ from that of the union of its constituent grid cells. We tackle these challenges with a multimodal and multitask approach, incorporating rich visual cues and graph context.

We propose *MAGRE* – a novel multitask and multimodal adaptive geospatial region embedding approach. In contrast to conventional methods, *MAGRE* partitions the geospatial region into smaller hexagonal grid cells, which can be flexibly aggregated to match the specific ROI. In addition to features from various cross-modal sources such as Points of Interest (POIs) and mobility data, *MAGRE* also extracts the image for each grid cell from OpenStreetMap (OSM)[1] to obtain visual information for creating comprehensive embeddings. We generate various graphs utilizing the extracted features. These graphs capture similarities and

[1] OpenStreetMap: https://www.openstreetmap.org/. The OpenStreetMap name is a trademark of the OpenStreetMap Foundation and is used with their permission. We are not endorsed by or affiliated with the OpenStreetMap Foundation.

rich context of grid cells based on various factors, including mobility patterns, locality, and infrastructural attributes. To train *MAGRE*, we propose a multi-task learning approach, which enables the embedding to learn region semantics from different perspectives and reduces overfitting through shared representations. Based on the fine-grained grid embeddings, *MAGRE* can efficiently generate embeddings for any ROI by embedding aggregation. The aggregated region embeddings effectively preserve the semantic information, as demonstrated in our experiments on several downstream tasks. Our contributions are as follows:

- We propose *MAGRE* – an adaptive region embedding approach, which creates representations that accurately capture the spatial properties and relationships between the hexagonal grid cells and can embed regions of flexible shape and size through efficient aggregation.
- *MAGRE* leverages multimodal data from various sources to build region embeddings. To the best of our knowledge, we are the first to incorporate visual cues from map images into region embedding, effectively capturing the context and features of urban regions. Our feature analysis results demonstrate the importance of map images across different tasks.
- To enhance embedding generalizability, *MAGRE* embraces multitask learning, where we train our model on two tasks and test the geospatial embeddings on unseen tasks. Experimental results demonstrate that *MAGRE* outperforms the state-of-the-art methods, leading to a root mean squared error reduction of 25.73% and 19.08% for crime rate prediction and check-in count prediction, respectively.

2 Definitions and Problem Formulation

In this section, we introduce the relevant definitions and formulate the problem of spatial region embeddings.

Definition 1 (Geospatial grid cell). *A geospatial grid cell, denoted as g, is a minimal spatial unit characterized by specific geometric boundaries. A grid cell is associated with features in different categories. Features of a grid cell g_i based on a feature category f are denoted as a vector $\vec{h}_f^i$.*

We adopt hexagonal geospatial grid cells. Feature category f can represent the frequency of different types of POIs or mobility patterns. We represent the relationships between the grid cells according to f as a grid graph.

Definition 2 (Grid graph). *We denote a grid graph as $\mathcal{G}_f = (\mathcal{V}, \mathcal{E}, \mathcal{A}_f)$, where $\mathcal{V} = \{g_1, ..., g_n\}$ represents the set of grid cells, $\mathcal{E}$ is the set of edges. $\mathcal{A}_f$ denotes the weighted adjacency matrix associated with the feature category f. $A_f^{ij} = sim(\vec{h}_f^i, \vec{h}_f^j)$, where $sim(\cdot)$ denotes the similarity function.*

Grid cell similarity can be computed as the cosine similarity between their feature vectors. To enable efficient grid cell representation, we rely on embeddings.

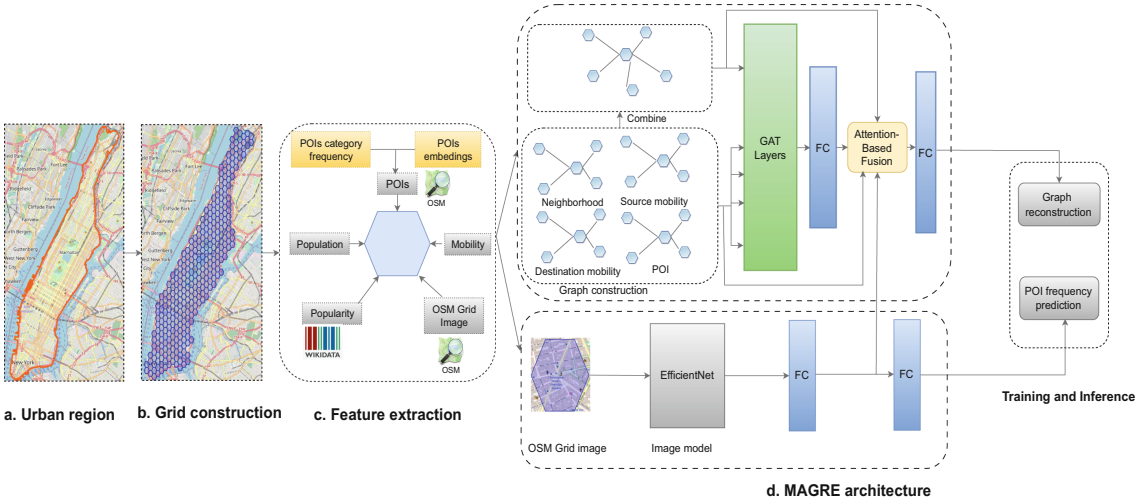


Fig. 2. The overall architecture of the proposed *MAGRE* approach. Map data: ©OpenStreetMap contributors, ODbL.

Definition 3 (Grid cell embedding). *For a grid cell g_i, an embedding $e_i = \phi(g_i)$, $e_i \in R^d$ is a d-dimensional dense vector representation of g_i. $\phi(\cdot)$ is an embedding function capturing semantic and contextual information of g_i.*

In this paper, we address the problem of adaptive spatial region embedding for spatial regions of user interest, e.g., a district or a new business area.

Definition 4 (Spatial region). *A spatial region, denoted as r, is a geographic area defined by specific boundaries. A spatial region r can be represented by a set of spatial grid cells $\{g_1, ..., g_n\}$ it contains or intersects with.*

We aim at generating embeddings tailored to any geospatial region, based on the embeddings of the contained spatial grid cells.

Definition 5 (Spatial region embedding). *For a given spatial region r, the geospatial embedding e_r is constituted by the aggregation of the grid cell embeddings $e_r = \gamma(\{\phi(g_i)\})$, where $g_i \in r$ and $\gamma(\cdot)$ is an aggregation function, such that e_r retains the semantics of r.*

3 The *MAGRE* Approach

The architecture of the proposed *MAGRE* approach is illustrated in Fig. 2. In this section, we provide a detailed description of each step.

3.1 Grid Construction and Feature Extraction

In this step, we partition the entire geographic area into hexagonal grid cells and extract features from multimodal data for each grid cell.

Grid Construction. We opt for a hexagonal grid to partition urban regions compared to other geometric shapes, as illustrated in Fig. 2b. The hexagonal grid

has several advantages. First, hexagonal grids experience less distortion caused by the earth's curvature compared to the shape of a fishnet grid [3]. Second, due to the consistent length of each side, the centroids of neighboring cells are equidistant [3]. We set the size of each side of the hexagonal grid to 250 m, such that the resulting hexagon area is comparable to [11].

Feature Extraction. We express each grid cell as feature vectors representing POIs count, mobility patterns, OSM images, population statistics, and popularity count, as illustrated in Fig. 2c.

- **POIs.** The type distribution of POIs in a region provides important semantic indicators, such as urban types. To capture such semantics, we extract all the POIs corresponding to each grid cell and map them to OSM categories, resulting in 12 categories. Each grid cell contains a feature vector of length 12 representing the *POIs categories frequency* with categories as amenity, barrier, highway, leisure, man-made, natural, office, power, public transport, railway, shop, and tourism. Additionally, to maintain POI semantics, we aggregate the names of all POI venues within a particular grid cell and utilize the sentence transformer [7] to generate *POIs embeddings*. The concatenated feature vector of *POIs category frequency* along with *POIs embeddings* is denoted $\vec{h}_{poi}$.
- **Mobility patterns.** Human mobility patterns are pivotal in understanding the underlying correlations between regions [10]. Regions with similar incoming or outgoing mobility patterns often have similar functions and are closely connected from the human mobility perspective [14]. The number of trips originating from and ending at a grid cell is concatenated, denoted as $\vec{h}_{mob}$.
- **Population and popularity.** A region's population can reflect socioeconomic indicators. We aggregate population statistics as a grid feature. We extract the popularity count of each grid cell using POI Wikidata links. A higher number of POI Wikidata links in a grid cell acts as a proxy for popularity. We denote the population and popularity frequency as $\vec{h}_{pp}$.
- **Map images.** The visual representation of spatial regions helps to recognize and distinguish various characteristics. For example, OSM distinct colors to represent different objects facilitate visual map interpretation. We partition the map into multiple images, each capturing a specific grid cell. These images can reveal substantial patterns, such as the POI density.

3.2 *MAGRE* Model Architecture

In this section, we present our model architecture in detail. We consider two tasks to design the objective functions: grid graph reconstruction, and POI frequency prediction. To learn the joint multitask representation, we apply an attention-based fusion, followed by training and inference, as illustrated in Fig. 2d.

Grid Graph Reconstruction. We first construct different grid graphs, capturing the semantic and spatial similarity between grids. Each graph $\mathcal{G}_f$ is constructed by computing an adjacency matrix $\mathcal{A}_f$ of all grid cells as described

in Definition 2. This results in a grid-graph $\mathcal{G}_{poi}$ based on POIs ($\vec{h}_{poi}$), two grid-graphs based on mobility patterns using source and destination frequency ($\vec{h}_{mob}$), represented as $\mathcal{G}_{src}$ and $\mathcal{G}_{dst}$, respectively. In addition, we create a grid graph $\mathcal{G}_{nbh}$ for the neighborhood information, capturing the relationship between grids based on their geospatial proximity. We also build a grid graph, which is a combination of the average of all the adjacency matrices from different grid graphs, represented as $\mathcal{G}_{cmb} = (\mathcal{V}, \mathcal{A}_{cmb})$, such that $\mathcal{A}_{cmb} = \frac{1}{|f|}\sum_{i=1}^{|f|}\mathcal{A}_i$, where $f \in \{poi, src, dst, nbh\}$, as illustrated in Fig. 2d. The intuition behind averaging adjacency matrices is that grid cells with high average weights in the combined matrix indicate strong and consistent connections across different modalities.

We employ Graph Attention Networks (GAT) [9] to extract meaningful representations from grid graphs. GAT is specially designed for graph-structured data and employs an attention mechanism. This mechanism facilitates the update of grid representations by efficiently propagating information to neighboring grids within each grid graph. We represent grid cell feature as $\vec{h^i}$ where $\vec{h^i} \in \{\vec{h}^i_{poi}||\vec{h}^i_{mob}||\vec{h}^i_{pp}\}$ and $||$ represents the concatenation operator. The GAT layer updates the grid representations through the following steps. First, we incorporate edge weights A^{ij} as an additional feature along with the grid features $\vec{h^i}$ and $\vec{h^j}$ in the learning process, given as $c_{ij} = \exp(\text{ReLU}(\vec{a}^T[W\vec{h^i}||W\vec{h^j}||W_eA^{ij}]))$. The c_{ij} calculation is performed only for grids $j \in N_i$, where N_i denotes the set of the top N neighbors of the grid i, ranked according to adjacency matrix weights for the grid i (including i). To ensure the comparability of coefficients across different grids, we normalize them using the softmax function $\alpha_{ij} = \text{softmax}(c_{ij})$. Next, we compute the updated grid representation $\vec{h^{i'}}$ by applying a weighted sum of the neighboring grid representations as $\vec{h^{i'}} = \sigma(\sum_{j\in N_i}\alpha_{ij}W\vec{h^j})$, where the weights are given by α_{ij} and σ denotes the activation function. To improve model convergence, we implement the skip connection mechanism, wherein certain layers in the neural network are skipped, and the output of one layer is directly fed to the subsequent layers. We concatenate the feature vector $\vec{h^i}$ with $\vec{h^{i'}}$, resulting in $\vec{h^{i''}}$ which denotes the representation of the grid cell i for a grid graph. We utilize a multi-head attention mechanism within each GAT layer to enhance performance, as proposed by [9]. In practice, we apply three GAT layers [9] followed by fully connected (FC) layers on each grid graph, namely $\mathcal{G}_{src}$, $\mathcal{G}_{dst}$, $\mathcal{G}_{poi}$, $\mathcal{G}_{nbh}$ and $\mathcal{G}_{cmb}$. This process yields the hidden representations $E_G = \{\vec{h}''_{src}, \vec{h}''_{dst}, \vec{h}''_{poi}, \vec{h}''_{nbh}, \vec{h}''_{cmb}\}$.

POI Frequency Prediction. In this task, we leverage Convolutional Neural Networks (CNN) to extract meaningful representations of grid images. We aim to train a model based on POI frequency, capable of learning object distribution within a grid image. We develop a regression model ψ incorporating the EfficientNet architecture [8] as its base model. EfficientNet is an image classification model known for its state-of-the-art accuracy, achieved with fewer model parameters. We customized EfficientNet for our regression task, which predicts the number of POIs in a given grid image. Formally, given a grid image dataset I_{img}, where $I_{img} \in \{i_1, .., i_k, .., i_n\}$ such that i_k represents the image for a given

grid cell k. We apply the regression model ψ which predicts the POI frequency $\hat{y}$. We extract the intermediate representation of the model, i.e., E_{img}.

Attention-Based Fusion. Finally, a multi-head attention-based fusion is applied to the embeddings from the multiple grid graphs and the grid images. This fusion helps to propagate knowledge across the representations of different modalities, given as $E = MultiHeadAtt(E_G||E_{img})$. To reduce the dimensionality, we apply an FC layer, i.e., $e = FC(E)$, representing the grid cell embeddings.

Training and Inference. We employ the graph reconstruction task to train the graph reconstruction module in an unsupervised way. That is, having obtained the different representations for each grid graph, we reconstruct the original adjacency matrix $\mathcal{A}_f$ with $\hat{\mathcal{A}}_f = sigmoid(e.e^T)$. We employ Mean Square Error (MSE) loss to compute the reconstruction loss, represented as $\mathcal{L}_f^{rec} = ||\mathcal{A}_f - \hat{\mathcal{A}}_f||^2$. The smooth L1 loss [5] is utilized as the loss function for predicting POI frequency in grid images which combines the benefits of both L1 and L2 loss, making it suitable for handling outlier values. For instance, the contrasting frequency of POIs between the Manhattan Central Park grid, which has very few POIs, and other regions illustrates this variability. The formal definition of the smooth L1 loss between the original POIs count (y) and the predicted values ($\hat{y}$) is computed as in [5]:

$$\mathcal{L}_{img}^{smooth} = \begin{cases} \sum_{i=1}^{n} 0.5(\hat{y}_n - y_n)^2/\beta & \text{if } |\hat{y}_n - y_n| < \beta \\ \sum_{i=1}^{n} |\hat{y}_n - y_n| - 0.5 * \beta & \text{otherwise,} \end{cases} \tag{1}$$

where β specifies the threshold for switching between L1 and L2 loss.

We define the loss function for our model as a combination of the loss of the two objective tasks: $\mathcal{L}_{tot} = \sum_{k \in f} \mathcal{L}_k^{rec} + \mathcal{L}_{img}^{smooth}$. During training, model parameters and all the embeddings are learned through backpropagation.

3.3 Embedding Aggregation for Spatial Regions

Once the grid cell embeddings are generated, the next task is to aggregate embeddings for a given region r. We sum the grid cell embeddings for a region r, represented by the set of spatial grid cells it contains or intersects with, and obtain $e_r = \sum_{i=1}^{m} e_i$, where m is the number of grid cells in the region r. Then, we utilize embedding e_r for the downstream tasks. Following [16], for the regression tasks, the aggregated embeddings are passed through a fully connected neural network, followed by the prediction layer resulting in the prediction value $\hat{o}_i$. To optimize the regression tasks, we use the MSE loss function as $\mathcal{L}_{agg} = \frac{1}{p} \sum_{i=1}^{p} (\hat{o}_i - o_i)^2$, where p is the number of spatial regions in the downstream task.

4 Experimental Setup

We assess the generalizability and effectiveness of the spatial region embeddings generated by *MAGRE* on unseen tasks. This section describes the downstream tasks, datasets, baselines, parameter settings, and evaluation metrics.

Downstream Tasks and Datasets. We experiment with three distinct downstream tasks, including two regression tasks – crime rate prediction and check-in count prediction, and one classification task – land use classification. For evaluation, we consider the following datasets containing publicly available statistical and geographical data:

- Crime rate statistics: We use crime statistics, i.e., the count of crimes per region, provided by [16].
- Check-in count statistics: We use the count of check-ins per region, provided by [16].
- Land use classification: We use the district divisions determined by the community boards [2] as the reference, which corresponds to 12 categories [16].

Following past works [12], we select the Manhattan City area. For feature generation, we consider the following data:

- POI data: approx. 48,000 POIs extracted from the OpenStreetMap.
- Taxi data: anonymized data which contains start and end locations of approximately 5 million taxi trips in 2015[2].
- Images: We extract images of each grid cell from OSM.
- Popularity data: We extract the Wikidata tag for each POI from OSM, and compute the number of Wikidata links for each POI with a SPARQL query.
- Population statistics for 3,930 administrative regions, aggregated to our grids (see footnote 2).

We map our hexagonal grids to the existing division of the Manhattan region, consisting of 180 census blocks based on street boundaries. This alignment ensures a fair and accurate comparison with the baselines. Particularly for the land classification task, we further cluster the aggregated embeddings into 12 groups, to align with the number of distinct labels in the ground truth [12].

Baselines. We compare our *MAGRE* method with the baseline methods for region embeddings. HREP [16] captures both intra-region and inter-region correlations by integrating statistical taxi data and POI data. MG-FN [12] is a joint learning approach, utilizing mobility patterns for region representation. MVURE [15] is a multi-view graph representation approach that uses region correlations based on POIs, taxi statistics, and check-in statistics to learn urban region embeddings. Hex2Vec [11] relies only on OSM data incorporating a skip-gram model to create vector representations of hexagonal regions. Similarly, RegionDCL [6] considers only OSM building footprints and employs dual contrastive learning for region embeddings. MV-PN [4] constructs a multi-view POI-POI network using POIs and human mobility data based on autoencoder. Moreover, we explore variations of *MAGRE* where we employ a Graph Autoencoder (GAE) and node2vec instead of GAT layers on the grid graphs, represented as $MAGRE_{n2v}$ and $MAGRE_{gae}$, respectively.

[2] https://opendata.cityofnewyork.us/.

Table 1. Performance of *MAGRE* and baselines on different tasks. %improv. shows the percentage improvement of *MAGRE* over the best baseline result.

Methods	Crime rate prediction			Check-in count prediction			Land use	
	MAE↓	RMSE↓	R^2 ↑	MAE↓	RMSE↓	R^2 ↑	NMI↑	ARI↑
RegionDCL	118.31	156.45	0.18	464.41	732.73	0.12	0.43	0.17
Hex2Vec	109.31	144.02	0.05	400.78	651.47	0.37	0.39	0.13
MV-PN	93.14	125.27	0.28	476.12	783.12	0.07	0.39	0.15
MVURE	65.41	91.63	0.61	297.12	494.36	0.63	**0.75**	0.55
MG-FN	77.34	98.32	0.56	321.44	510.04	0.61	0.74	0.55
HREP	66.66	85.13	0.67	273.27	411.98	0.75	**0.75**	0.53
$MAGRE_{n2v}$	60.46	82.99	0.69	302.35	483.93	0.65	0.66	0.42
$MAGRE_{gae}$	86.45	118.91	0.35	297.84	507.52	0.61	0.20	0.03
MAGRE	**35.47**	**63.22**	**0.82**	**209.39**	**333.34**	**0.83**	**0.75**	**0.57**
% improv	45.77	25.73	22.38	23.37	19.08	10.66	0.0	3.63

Evaluation Metrics. In the regression tasks, we compute all the metrics using 5-fold cross-validation with a train-test split of 80-20%. We utilize Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) to evaluate the performance of the prediction models. Furthermore, to measure how well the regression model fits the observed data, we use the coefficient of determination, denoted by R^2. To assess the quality of the clustering results, we utilize Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI).

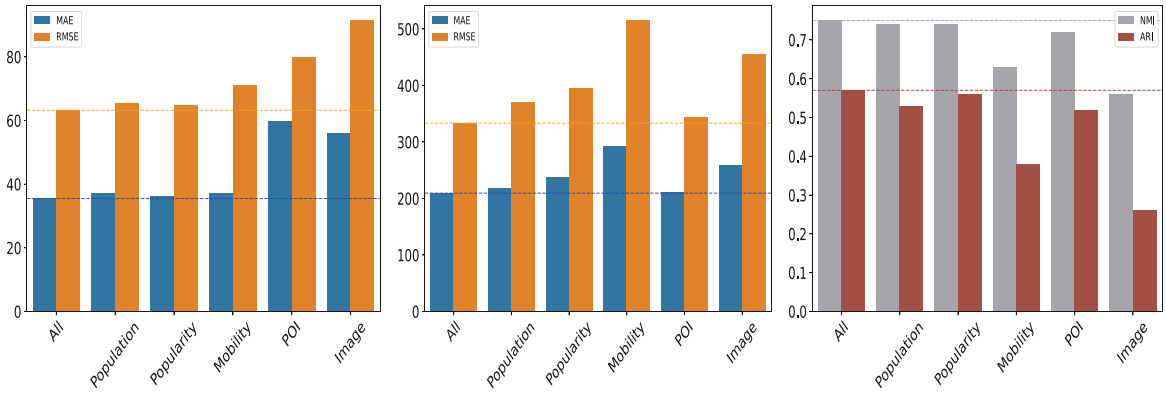
Parameter Settings. For the hexagonal grid, the number of neighbors (N_i) for each grid cell is seven, including the grid itself. For the grid embedding, we chose an embedding dimension of 128 and utilized the Adam optimizer with a learning rate of 0.001 [16]. For L1 smooth loss, β is set to its default value 1. *MAGRE* is trained for 2000 epochs. For downstream tasks, we adopt the hyperparameters as in [16].

5 Evaluation Results

In this section, we present evaluation results of the *MAGRE* compared to baselines and analyze the importance of features.

General Performance. First, we discuss the overall performance of our approach on different downstream tasks. In regression tasks, *MAGRE* outperforms the baseline methods in both MAE, RMSE, and R^2, as illustrated in Table 1. HREP, being the best-performing baseline in both regression tasks regarding RMSE, demonstrates its effectiveness with prompt learning, which replaces the direct use of region embedding in the downstream tasks. RegionDCL's poor performance is attributed to its sole reliance on OSM building data for region embeddings, failing to capture sufficient semantics for effective prediction on

downstream tasks. Comparing the variations of our model, we observe a similar trend as reported in [16], i.e., $MAGRE_{n2v}$ outperforms $MAGRE_{gae}$. For the land use classification, as can be seen in Table 1, *MAGRE* achieves the highest scores in terms of ARI and surpasses the best-performing baselines by 3.63%. Regarding NMI, *MAGRE* achieves comparable performance with the best-performing baselines MVURE and HREP. The overall evaluation results on all three tasks demonstrate the effectiveness of the *MAGRE*'s embeddings.



(a) crime rate prediction (b) check-in count prediction (c) land use classification

Fig. 3. Results of removing *MAGRE* features one at a time.

Feature Analysis. To analyze the impact of each feature category on the model performance, we systematically remove one feature category at a time, as shown in Fig. 3. We observe that the best results in all three tasks are obtained by using all features, demonstrating the effectiveness of our model in capturing the region semantics. Removing the OSM image leads to a notable increase in MAE and RMSE in regression tasks (Fig. 3a and 3b) and a decrease in NMI and ARI for land use classification (Fig. 3c). Specifically, this leads to a 44.62% increase in RMSE for crime rate prediction, a 36.67% increase in RMSE for check-in count prediction, and a 54.38% decrease in NMI. This finding emphasizes that OSM images contain useful information for learning region embeddings. The absence of the mobility feature negatively affects check-in count prediction, which is intuitive given the close relationship between check-in statistics and mobility patterns. Furthermore, POI features play a crucial role in crime rate prediction.

6 Case Study: Crime Rate Prediction on ROIs

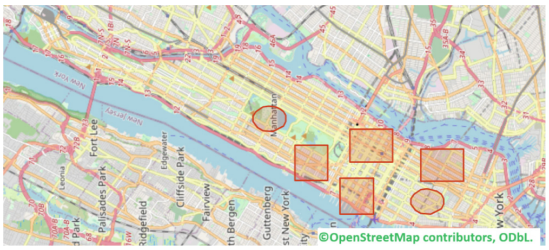


Fig. 4. Example of different ROI shapes.

Table 2. Crime rate prediction on ROIs.

Methods	MAE↓	RMSE↓
HREP	82.81	113.50
MVURE	113.40	166.41
MAGRE	**30.13**	**54.45**

Our case study is designed to showcase the adaptability and flexibility of *MAGRE*. To demonstrate its capabilities, we predict crime rates in ROIs with varying sizes and shapes. We randomly chose 200 locations within the Manhattan boundary. At each of these selected locations, we employ a randomization process to generate one of three distinct shapes for spatial regions: square, rectangle, or circle, as illustrated in Fig. 4. The area of each shape is randomly generated, with the upper bound of the area of the largest administrative region of Manhattan. Subsequently, we aggregate the grid embeddings to create region embeddings for these 200 ROIs, following the steps in Sect. 3.3. For baseline methods, we chose the top-2 best-performing baselines, i.e., HREP and MVURE. We conduct a five-fold cross-validation to obtain prediction results. As baseline methods can only compute predictions for administrative regions, we compute a weighted sum of the administrative region's prediction scores for a fair comparison. The weights are determined by the proportion of the overlap of administrative regions within the ROIs. As shown in Table 2, *MAGRE* outperforms the selected baseline methods, leading to an MAE reduction of 63.61% and a RMSE reduction of 52.02% as compared to the best-performing baseline, i.e., HREP. The gap in crime rate prediction scores between the ROIs and administrative boundaries (Table 1 and Table 2) for the two baseline methods indicates a lack of adaptability in these approaches. These outcomes highlight the adaptability and effectiveness of *MAGRE* in handling varying ROIs.

7 Related Work

This section briefly summarizes the related works in the representation learning of geospatial regions. Some recent works incorporate POIs and mobility data to construct meaningful region embeddings for fixed administrative boundaries. For instance, Zhang et al. [15] introduced a multi-view graph representation approach, which considered POI data and mobility patterns to generate a representation of fixed-size urban regions. Similarly, Zhou et al. [16] utilized a prompt learning method by leveraging both POI and mobility data. Wu et al. [12] focused on leveraging mobility data alone and trained for a specific task, i.e., mobility distribution. Xi et al. [13] integrated the satellite imagery alongside POI data.

Fu et al. [4] built a multi-view POI-POI network utilizing POI data and employed an autoencoder for region embedding. Woźniak et al. [11] relied only on OSM data incorporating a skip-gram model, generating vector representations for each hexagonal region. Similarly, Li et al. [6] utilized only OSM building footprints for region representation with dual contrastive learning. The state-of-the-art approaches either create region embeddings for fixed administrative boundaries or rely on limited data sources. With *MAGRE*, we acquire adaptive latent representations of grid cells that can be flexibly aggregated to a spatial region of any shape and size.

8 Conclusion

We proposed *MAGRE* – a novel approach that leverages multitask learning and multimodal spatial embeddings to create an adaptive representation of urban regions. *MAGRE* leverages fine-grained hexagonal grid cells, enabling a more precise and detailed depiction of their spatial characteristics. *MAGRE* can efficiently generate embeddings of any ROI by embedding aggregation and effectively preserves the semantics, as demonstrated in our experiments. Experimental results on three downstream tasks demonstrate that *MAGRE* exhibits superior performance compared to baseline methods, highlighting the benefits of multitasking and multimodal approach for learning latent representations of urban regions.

Acknowledgements:. This work was partially funded by the Federal Ministry for Economic Affairs and Climate Action (BMWK), Germany ("ATTENTION!", 01MJ22012C).

References

1. Bartunov, S., Kondrashkin, D., Osokin, A., Vetrov, D.P.: Breaking sticks and ambiguities with adaptive skip-gram. In: AISTATS 2016. JMLR.org (2016)
2. Berg, B.F.: New York City Politics: Governing Gotham. Rutgers University Press, New Brunswick (2007)
3. Birch, C.P., Oom, S.P., Beecham, J.A.: Rectangular and hexagonal grids used for observation, experiment and simulation in ecology. Ecol. Model. **206**(3–4), 347–359 (2007)
4. Fu, Y., Wang, P., Du, J., Wu, L., Li, X.: Efficient region embedding with multi-view spatial networks: a perspective of locality-constrained spatial autocorrelations. In: AAAI 2019, pp. 906–913. AAAI Press (2019)
5. Girshick, R.B.: Fast R-CNN. In: IEEE, ICCV 2015, pp. 1440–1448. IEEE Computer Society (2015)
6. Li, Y., Huang, W., Cong, G., Wang, H., Wang, Z.: Urban region representation learning with openstreetmap building footprints. In: ACM SIGKDD. ACM (2023)
7. Reimers, N., Gurevych, I.: Sentence-bert: sentence embeddings using siamese bert-networks. In: EMNLP-IJCNLP 2019. ACL (2019)
8. Tan, M., Le, Q.V.: Efficientnet: rethinking model scaling for convolutional neural networks. In: ICML 2019, pp. 6105–6114. PMLR (2019)

9. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: ICLR 2018. OpenReview.net (2018)
10. Wang, H., Li, Z.: Region representation learning via mobility flow. In: ACM, CIKM 2017, pp. 237–246. ACM (2017)
11. Wozniak, S., Szymanski, P.: hex2vec: context-aware embedding H3 hexagons with openstreetmap tags. In: GeoAI@SIGSPATIAL 2021, pp. 61–71. ACM (2021)
12. Wu, S., et al.: Multi-graph fusion networks for urban region embedding. In: IJCAI 2022 (2022)
13. Xi, Y., Li, T., Wang, H., Li, Y., Tarkoma, S., Hui, P.: Beyond the first law of geography: learning representations of satellite imagery by leveraging point-of-interests. In: WWW 2022, pp. 3308–3316. ACM (2022)
14. Yao, Z., Fu, Y., Liu, B., Hu, W., Xiong, H.: Representing urban functions through zone embedding with human mobility patterns. In: IJCAI 2018 (2018)
15. Zhang, M., Li, T., Li, Y., Hui, P.: Multi-view joint graph representation learning for urban region embedding. In: IJCAI 2020, pp. 4431–4437. ijcai.org (2020)
16. Zhou, S., He, D., Chen, L., Shang, S., Han, P.: Heterogeneous region embedding with prompt learning. In: AAAI 2023. AAAI Press (2023)

Appendix C

Publication: Towards Effective, Robust and Utility-preserving Watermarking of GPS Trajectories

Rajjat Dadwal, Thorben Funke, Michael Nüsken, and Elena Demidova

ACM Transactions on Spatial Algorithms and Systems, 2024

DOI: 10.1145/3701558

Rajjat Dadwal, Thorben Funke, Michael Nüsken, and Elena Demidova, "Towards effective, robust, and utility-preserving watermarking of GPS trajectories", accepted for publications in ACM Transactions on Spatial Algorithms and Systems (TSAS), accepted on 03 October 2024. The original version of the record can be found at: https://dl.acm.org/doi/10.1145/3701558.

Towards effective, robust and utility-preserving watermarking of GPS trajectories

RAJJAT DADWAL, Data Science & Intelligent Systems Group (DSIS), University of Bonn, Lamarr Institute for Machine Learning and Artificial Intelligence, Germany
THORBEN FUNKE, L3S Research Center, Leibniz Universität Hannover, Germany
MICHAEL NÜSKEN, Bonn-Aachen International Center for Information Technology, Germany
ELENA DEMIDOVA, Data Science & Intelligent Systems Group (DSIS), University of Bonn, Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

Personal GPS trajectory is essential for businesses and emerging data markets due to its relevance in various data-driven methods, including traffic forecasting, accident prediction, and profiling driving behavior. Watermarking is a method that facilitates verification of data ownership and authenticity by embedding provenance information into the data. Whereas watermarking is commonly adopted in the image and audio domains, only a few initial watermarking methods exist for GPS trajectory data. GPS trajectory watermarking is particularly challenging due to the spatio-temporal data properties and easiness of data modification. As a result, existing watermarking methods often embed only minimal provenance information, lack robustness, and can fail to preserve data utility for downstream applications. In this work, we propose *W-Trace* - a novel, effective, robust, and utility-preserving GPS trajectory watermarking method. *W-Trace* transforms a GPS trajectory into a complex domain and applies the Fourier transformation to decompose the trajectory into the frequency representation. *W-Trace* embeds watermarks in the frequency representation and verifies them in a spatiotemporally-aware procedure. We demonstrate the effectiveness, robustness, and utility of the proposed *W-Trace* approach in realistic settings using real-world GPS trajectory datasets. In contrast to the baselines, the proposed *W-Trace* approach is robust to a wide range of trajectory modifications while preserving the GPS trajectory characteristics required for the downstream applications.

CCS Concepts: • **Information systems** → **Spatial-temporal systems**; • **Security and privacy**;

Additional Key Words and Phrases: GPS trajectory, Watermarking, Data provenance

1 INTRODUCTION

Personal GPS trajectory data originating from vehicle sensors, navigation devices, and mobile apps is critical for a wide range of real-world applications. These applications encompass traffic forecasting, accident anticipation, route optimization, and profiling driving behavior for insurance-related purposes [9, 16, 18, 39]. GPS trajectory data can contain personal information, including visited locations, travel routes, and driver profiles [11, 14, 16]. Applications that depend on personal GPS trajectory data necessitate reliable approaches to confirm data provenance and authenticity. The verification of data provenance is becoming increasingly important in the context of confirming compliance with the consent requirements for personal data processing according to The General Data Protection Regulation (GDPR) in the European Union. This verification can assist authorized

Authors' addresses: Rajjat Dadwal, dadwal@cs.uni-bonn.de, Data Science & Intelligent Systems Group (DSIS), University of Bonn, and Lamarr Institute for Machine Learning and Artificial Intelligence, Bonn, Germany; Thorben Funke, tfunke@L3S.de, L3S Research Center, Leibniz Universität Hannover, Hannover, Germany; Michael Nüsken, nuesken@bit.uni-bonn.de, Bonn-Aachen International Center for Information Technology, Bonn, Germany; Elena Demidova, elena.demidova@cs.uni-bonn.de, Data Science & Intelligent Systems Group (DSIS), University of Bonn, and Lamarr Institute for Machine Learning and Artificial Intelligence, Bonn, Germany.

data processors in validating consent for data processing and also help detect unauthorized sharing of personal trajectory data. In addition, effective GPS trajectory provenance verification is becoming increasingly important in the insurance industry. This verification can help confirm driver identity during risk assessment for personalized insurance policies (e.g., [25]) and for validating insurance claims.

Digital watermarking, which refers to techniques that embed provenance information (a watermark) within noise-tolerant data, can facilitate verification of the provenance and authenticity of GPS data. Extensive research in watermarking has predominantly focused on the media domain, particularly in protecting images [27, 43], audio files [2, 21], and videos [26, 28]. Notable examples of perceptible watermarks include logos embedded in the images, a feature commonly found in photo-sharing platforms. Another common application involves audio files, where imperceptible watermarks aid in tracking the file origin on illicit sharing platforms [13]. Nevertheless, digital watermarking personal GPS trajectories has received only relatively limited attention in research (e.g., [19], [32]).

Digital watermarking GPS trajectories poses several challenges. Embedding a watermark in the GPS trajectory data is subject to a trade-off. On the one hand, a watermark should be effective and robust, i.e., embed sufficient information for verification and be strong enough not to be modified or removed by potential adversaries. On the other hand, the watermark impact on the data should be minimal to preserve data utility for downstream applications. In addition to this general challenge for digital watermarking, real-world GPS trajectories pose unique challenges due to their non-uniform sampling rates and positional inaccuracy, making them vulnerable to various modification attacks aiming to remove the watermark, such as point removal, point addition, and resampling along the path [32]. State-of-the-art watermarking methods in the trajectory domain are either ineffective, i.e., they can only embed a small amount of provenance information [32], or lack robustness [19]. Furthermore, the utility of the watermarked trajectories in downstream tasks is not sufficiently studied in previous works [19, 32].

In this article, we present *W-Trace* – a novel approach that enables effective, robust, and utility-preserving watermarking of GPS trajectories. The *W-Trace* approach operates as follows. First, *W-Trace* splits the trajectories into sub-trajectories and represents two-dimensional trajectory coordinates as complex numbers. Following that, *W-Trace* applies a Discrete Fourier Transform (DFT) to each sub-trajectory and inserts an imperceptible watermark into the Fourier descriptors. *W-Trace* incorporates a watermark into each frequency descriptor, enabling watermark dispersion across the frequency components, thereby increasing the amount of embedded information. Furthermore, splitting trajectories enables different watermarks to be inserted into the sub-trajectory segments. Embedding watermarks in the DFT domain enhances the watermarking robustness concerning scaling, translation, and resistance to noise [7]. Furthermore, *W-Trace* controls the amount of modification introduced by watermarking to maintain the trajectory utility. *W-Trace* effectively adjusts to the real-world GPS trajectory data properties, including sampling rate and length variations.

In summary, our contributions are as follows:

(1) We propose *W-Trace* — a novel watermarking approach for GPS trajectories that represents two-dimensional trajectory coordinates as complex numbers and adopts DFT to enable effective, robust, and utility-preserving watermark embedding into the frequency domain.
(2) *W-Trace* facilitates embedding more information into the watermark, compared to the state-of-the-art methods, by spreading the watermark throughout the entire trajectory, resulting in a more effective and robust approach.

Table 1. Notation table.

Notation	Meaning
B	boolean value
c	complex numbers ($\mathbb{C}$)
d_n	number of trajectories in the dataset
i	imaginary unit
p	GPS point
s	watermark strength
T	GPS trajectory
$\widetilde{T}$	watermarked GPS trajectory
$\widetilde{T}'$	attacked GPS trajectory
w	watermark vector
ts	GPS timestamp
$D(.)$	distance function
$AT(.)$	attack function
$EMB(.)$	watermark embedding function
$M(.)$	predictive model
$VER(.)$	watermark verification function
α	amplitude
φ	phase angle
σ	modification threshold
θ	attack parameter

(3) *W-Trace* controls the amount of trajectory modification introduced by watermarking to preserve the essential trajectory characteristics and maintain the trajectory utility for real-world applications.

We extensively evaluate our proposed *W-Trace* approach using two real-world GPS trajectory datasets. Our evaluation results confirm the effectiveness and robustness of *W-Trace* under a comprehensive set of adversarial trajectory modifications, including noise addition, point replacement, and length modifications. We demonstrate that under the majority of the considered attacks, *W-Trace* retains the watermark with a success rate of 100%. Furthermore, we verify the utility of watermarked trajectories on several downstream applications, such as map matching and trajectory user linking. Our results confirm that the utility of watermarked trajectories from our approach is comparable to original trajectories in both downstream tasks. To enhance reproducibility and encourage further research, we make our algorithms accessible to the community as open-source software.[1]

This article builds upon and substantially extends our preliminary work [10]. This extended version provides a detailed formalization and method description, including watermark embedding and verification algorithms. Furthermore, we conduct extensive experiments to assess the robustness and utility of our proposed watermarking approach in downstream applications.

2 DEFINITIONS AND PROBLEM FORMULATION

In this section, we introduce the relevant definitions and the problem statement. Notations are summarized in Table 1.

[1]Software: https://github.com/Rajjat/watermarkingTrajectory

A GPS trajectory is a sequence of geospatial locations and associated timestamps. In the context of this work, a GPS trajectory represents the movement of a person or a vehicle.

Definition 2.1 (GPS Trajectory). A GPS trajectory T is a sequence of GPS points arranged in chronological order and associated with their respective timestamps:

$$T = [(p_j, ts_j)], \text{with } ts_j < ts_{j+1} \text{ for all } j,$$

where $p_j = (a_j, b_j)$ denotes the two-dimensional points with the latitude a_j and longitude b_j, and ts_j refers to the timestamp associated with p_j. The number of points in the trajectory is denoted as the trajectory length, $N = \ell en(T)$. We denote the number of trajectories in the data as d_n.

Real-life trajectories exhibit differing lengths. We split these trajectories into fixed-length sub-trajectories to facilitate effective and robust watermarking.

Definition 2.2 (Sub-trajectory). A trajectory T can be conceptualized as a chronological sequence comprising sub-trajectories of a fixed length m, represented as $T = [t_1, ..., t_n]$, where n is the number of sub-trajectories in the trajectory T and each sub-trajectory t_i has length m.

We embed a watermark into the trajectory to enable data provenance and authenticity verification. We represent a watermark as a vector. The dimensionality of the watermark vector corresponds to the sub-trajectory length.

Definition 2.3 (Watermark vector and watermark sequence). A watermark w is a vector with the dimensionality m. We denote the sequence of the watermark vectors embedded into the sub-trajectories within the trajectory $T = [t_1, ..., t_n]$ as $W = [w_1, ..., w_n]$, where w_i is a watermark vector of length m embedded into the corresponding sub-trajectory t_i.

Watermark embedding is a process of inserting watermark vectors into trajectory data.

Definition 2.4 (Watermark embedding). Given a GPS trajectory T, a watermark sequence W is embedded into T with an embedding function $\text{EMB}(\cdot)$,

$$\widetilde{T} = \text{EMB}(T, W),$$

where $\widetilde{T}$ is the watermarked trajectory.

Watermark verification is a process that assesses whether the given watermark sequence is embedded into the trajectory.

Definition 2.5 (Watermark verification). Given an original trajectory T, a watermark sequence W, and a GPS trajectory $\widetilde{T}$, the verification function

$$VER(T, W, \widetilde{T}, \theta_v) \rightarrow B, B \in \{true, fa\ell se\}$$

assesses whether the given watermark sequence W is embedded into the trajectory $\widetilde{T}$. θ_v are verification parameters that are approach-specific.

In an attempt to remove or destroy the watermark, an adversary can modify the watermarked trajectory $\widetilde{T}$. Such modification (also referred to as an attack) $\widetilde{T}' = AT(\widetilde{T}, \theta)$ on the watermarked trajectory $\widetilde{T}$, where θ represents the specific attack parameter, leads to a noised trajectory $\widetilde{T}'$.

Definition 2.6 (Attack). Given a watermarked GPS trajectory $\widetilde{T} = EMB(T, W)$, an attack $\widetilde{T}' = AT(\widetilde{T}, \theta)$ aims to prevent a watermark verification:

$$VER(T, W, \widetilde{T}, \theta_v) \rightarrow B, VER(T, W, \widetilde{T}', \theta_v) \rightarrow B', B' \not\equiv B.$$

Watermarking aims to enable effective watermark verification in the presence of noisy data. The watermarking approach is said to be robust against an attack $AT(\cdot)$ if the watermark verification function $VER(\cdot)$ outputs the same result for the watermarked trajectory $\widetilde{T}$ and the noised trajectory $\widetilde{T}'$ resulting from this attack.

Definition 2.7 (Robust watermarking). Given a watermarked GPS trajectory $\widetilde{T}$, an attack $\widetilde{T}' = AT(\widetilde{T}, \theta)$ and a watermark verification function $VER(\cdot)$, the watermarking is robust against $AT(\cdot)$ if $VER(\cdot)$ outputs equivalent labels for $\widetilde{T}$ and $\widetilde{T}'$:

$$VER(T, W, \widetilde{T}, \theta_v) \rightarrow B, VER(T, W, \widetilde{T}', \theta_v) \rightarrow B', B' \equiv B.$$

Trajectory modifications, including watermarking and attacks, can affect the utility of the trajectory data for real-world applications. Applications considered in this work include predictive tasks such as traffic forecasting, accident prediction, route planning, and profiling driving behavior. We formalize such applications as a predictive model $M(\cdot)$.

Definition 2.8 (Predictive model). Given a GPS trajectory T, a predictive model $M(\cdot)$ takes a trajectory T and parameters $param$ as input and outputs a label L, i.e.,

$$M(T, param) \rightarrow L.$$

Depending on the specific application, L can represent various categories, such as traffic speed, accident likelihood, and the risk category of the driver profile.

We refer to a trajectory modification as utility-preserving regarding $M(\cdot)$ if applying $M(\cdot)$ to the original and the modified trajectories results in the same label.

Definition 2.9 (Utility-preserving modification). Given a GPS trajectory T, and a predictive model $M(\cdot)$, the modification $\widetilde{T} = MOD(T, ...)$ is utility-preserving regarding $M(\cdot)$ if $M(\cdot)$ outputs equivalent labels for $\widetilde{T}$ and T:

$$M(T, param) \rightarrow L, M(\widetilde{T}, param) \rightarrow L', L' \equiv L.$$

Trajectory modifications, including watermarking and attacks, are subject to a trade-off. On the one hand, more substantial modifications are desirable for modification effectiveness. In the case of watermark embedding, larger modifications can be used to increase the information content of the watermark. In the case of an attack, inserting a larger amount of noise is more likely to destroy the watermark. On the other hand, larger modifications can reduce the utility of the data for real-world applications by changing the trajectory and its properties. Therefore, data transformations in a utility-preserving modification are limited.

From the practical perspective, analyzing all potential modifications, including watermarks and attacks, and their impact on the trajectory utility is infeasible. Intuitively, a trajectory that results from a utility-preserving modification should be similar to the original trajectory. The similarity can be measured using the distance between the trajectories. We capture this intuition with a modification threshold.

Definition 2.10 (Modification threshold). A modification threshold σ bounds a distance D between the trajectories.

$$\mathrm{D}(T, \widetilde{T}) \leq \sigma.$$

Given a modification threshold σ, we refer to $\widetilde{T}$ as a σ-modification of T if $\widetilde{T}$ results from a modification of T and the distance between these two trajectories is at most σ. We denote such modification as $MOD_\sigma(T)$.

We refer to [36] for a survey of trajectory distance measures. In this work, we adopt the Haversine distance that considers the curvature of the Earth's surface for geographic distance computation. In our experiments, we work with a modification threshold of 10 meters, which reflects the typical inaccuracy of GPS sensors [4]. We further discuss the practical impact of watermarking on real-world applications under this modification threshold in Section 6.2. Overall, the objective of robust and utility-preserving trajectory watermarking is to ensure that the verification function $VER(T, W, \widetilde{T}', \theta_v)$ can correctly verify the watermark in a modified trajectory $\widetilde{T}'$ if $\widetilde{T}'$ is a σ-modification of $\widetilde{T}$ in which the watermark sequence W is embedded:

$$\widetilde{T} = EMB(T, W), \widetilde{T}' = MOD_\sigma(\widetilde{T}), VER(T, W, \widetilde{T}, \theta_v) \rightarrow B, VER(T, W, \widetilde{T}', \theta_v) \rightarrow B' \Longrightarrow B' \equiv B.$$

3 THE *W-TRACE* APPROACH

This section presents our proposed watermark embedding and verification method (*W-Trace*). *W-Trace* aims at effective and robust watermarking, while preserving the utility of the GPS trajectory data. First, we present our watermarking approach that embeds a watermark into GPS trajectories in Section 3.1. Then, in Section 3.2, we present the watermark extraction and verification mechanism. Finally, in Section 3.3, we discuss the computational complexity of the proposed watermark embedding and verification algorithms. An overview of *W-Trace* is illustrated in Fig. 1.

3.1 Watermark Embedding

Watermark embedding aims to incorporate a watermark into a given GPS trajectory. According to Definition 2.4, the watermark embedding function $\widetilde{T} = EMB(T, W)$ takes the trajectory $T = [(p_j, ts_j)]$, $p_j = (a_j, b_j)$ and a watermark sequence W as an input and outputs the watermarked trajectory $\widetilde{T}$. First, each trajectory is split into sub-trajectories $T = [t_1, ..., t_n]$ of equal length, corresponding to the dimension of the watermark vectors $w \in W$. The dimension of the watermark vector is a parameter. We analyze the impact of this parameter in Section 6.4.2. The embedding function associates each GPS point $p_j = (a_j, b_j)$ in a sub-trajectory $t \in T$ with a complex number,

$$c_j = a_j + ib_j, \tag{1}$$

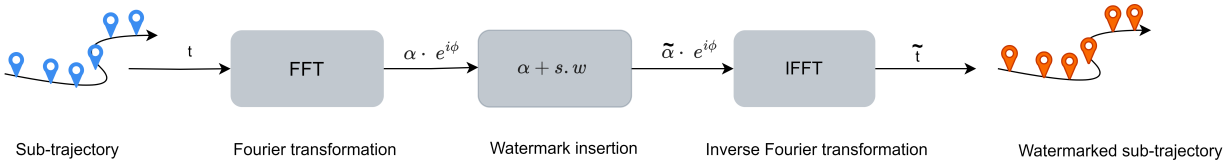
where i is the imaginary unit. The advantage of utilizing complex numbers is to spread the watermark into both coordinates at the same time. Next, we apply a Discrete Fourier Transform (DFT) [41] to each sub-trajectory, where we utilize the Fast Fourier Transform (FFT) [31] algorithm for efficiency. The Discrete Fourier Transform (DFT) retrieves a frequency domain representation of the input, which is a sequence of complex numbers with the same length as the input. The FFT algorithm takes the list of positions $c = [c_j]_{1 \le j \le m}$ from the sub-trajectory, represented as complex numbers, as input. We represent the resulting frequency representations using the amplitude α and phase angle φ:

$$\alpha \cdot e^{i\varphi} \leftarrow \mathrm{FFT}(c). \tag{2}$$

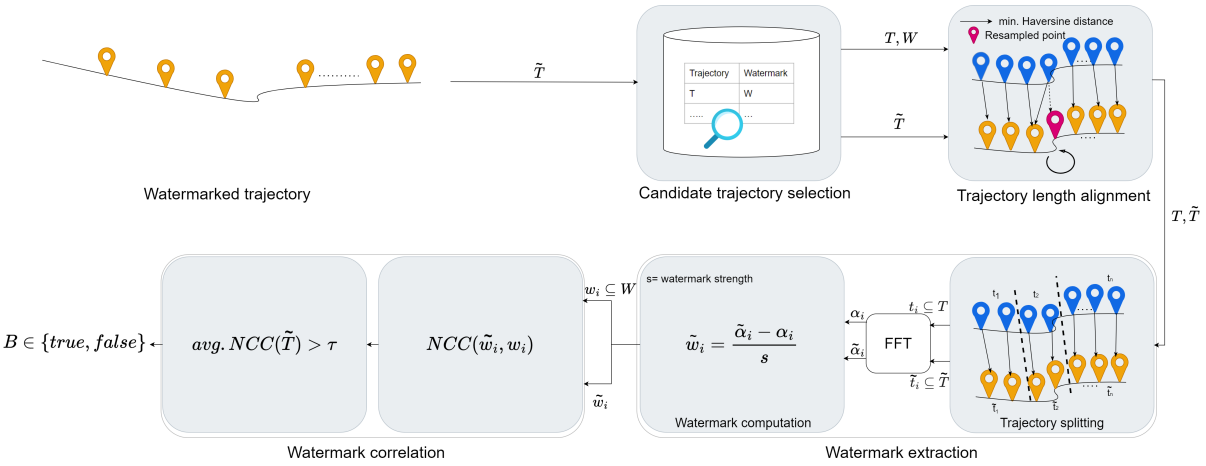
The watermark $w \in W$ with strength $s \in (0, 1)$ is inserted into the amplitude α of the sub-trajectory t:

$$\widetilde{\alpha} = \alpha + s \cdot w. \tag{3}$$

The watermarks may be uniform across all sub-trajectories or vary. In our approach, we represent the watermark w as a vector. Each element of the watermark vector is assigned a random value from 1, -1, and 0. This random assignment ensures that the watermark vector is distinct for each sub-trajectory. Restrictions on the values of watermark vectors and watermark strength $s \in (0, 1)$ help minimize substantial trajectory alterations and preserve the utility of watermarked trajectories.



(a) Watermark embedding in *W-Trace*.



(b) Watermark verification in *W-Trace*.

Fig. 1. Overview of the *W-Trace* approach.

The strength of a watermark is subject to a trade-off. On the one hand, a watermark should be effective and robust. That means that the watermark should be strong enough to enable verification and resist removal attempts by potential adversaries. A watermark with a higher strength contributes to more robust watermarking, making it more resistant to various attacks [37]. On the other hand, the watermark should have a minimal impact on the relevant data characteristics, preserving data utility. A watermark with a higher strength results in a more substantial modification to the trajectory and may affect the trajectory utility. The strength s of the watermark is a parameter of the proposed approach. We analyze the impact of the watermark strength in Section 6.4.1.

After the watermark is inserted into the amplitude of the trajectory, the subsequent step involves an inverse FFT (IFFT) to retrieve the watermarked trajectory:

$$\widetilde{t} = (\widetilde{a}, \widetilde{b}) \leftarrow \text{IFFT}(\widetilde{\alpha} \cdot \mathrm{e}^{i\varphi}), \tag{4}$$

where $\widetilde{t}$ is a watermarked sub-trajectory. We concatenate all the watermarked sub-trajectories into the watermarked trajectory $\widetilde{T}$. The watermark embedding procedure is summarized in Algorithm 1 and is illustrated in Fig. 1a.

To enable watermark verification, the user stores the watermark sequence. Verifying ownership of watermarked trajectory data can be accomplished by hashing the original data, the applied watermark sequence, and the watermarking parameters onto a distributed ledger, as suggested by Pan et al. [32]. It is important to note that such distributed ledger-based storage is not tied to any particular watermarking method and may not be required for all use cases.

3.2 Watermark Verification

Watermark verification is a process that assesses whether the given watermark sequence W is embedded into the trajectory $\widetilde{T}$. According to Definition 2.5, the verification function is defined

Algorithm 1 Watermark embedding

Input: GPS trajectory T, watermark sequence W.
Hyperparameters: watermark strength s, sub-trajectory length m.
Output: Watermarked GPS trajectory $\widetilde{T}$.

1: Split each trajectory T into fixed-length sub-trajectories $[t_1, ..., t_n]$.
2: **for** each sub-trajectory $t \in T$ **do**
3: Represent the sub-trajectory t as complex numbers c, see Eq. (1).
4: Apply FFT to c and represent the result in polar coordinates α, φ.
5: Add the watermark $w \in W$ to the amplitude α, see Eq. (3).
6: Apply inverse FFT to obtain the resulting watermarked sub-trajectory: $\widetilde{t} = \text{IFFT}(\widetilde{\alpha}, \varphi)$.
7: **end for**
8: Concatenate the watermarked sub-trajectories into $\widetilde{T}$.
9: **return** $\widetilde{T}$.

as: $VER(T, W, \widetilde{T}, \theta_v) \rightarrow B, B \in \{true, false\}$, where T is an original trajectory, W is the watermark sequence, and $\widetilde{T}$ is a GPS trajectory to be verified. In the context of this work, θ_v refers to the watermark strength parameter s adopted in the watermark embedding process.

The verification process includes four steps: selection of a candidate trajectory T, trajectory length alignment between T and $\widetilde{T}$, watermark extraction from $\widetilde{T}$, and watermark correlation. This process is summarized in Algorithm 2 and is illustrated in Fig. 1b.

Candidate trajectory selection. The watermark verification process requires the original trajectory T as input. As a candidate original trajectory T, we search for the closest original trajectory based on the minimum Haversine distance to $\widetilde{T}$. We evaluate this step in Section 6.4.4.

Trajectory length alignment. Our watermark verification process requires T and $\widetilde{T}$ to be of the same length. We align each point in T to a point in $\widetilde{T}$ based on the minimum Haversine distance. If the trajectory length of $\widetilde{T}$ is smaller than that of T, multiple points from T can be aligned to a point in $\widetilde{T}$. In this case, we add duplicate points to $\widetilde{T}$ to create a one-to-one alignment, as illustrated in Fig. 1b. If the length of $\widetilde{T}$ exceeds T, multiple points from $\widetilde{T}$ can be aligned to the point in T. In this case, we remove such points from $\widetilde{T}$ to create a one-to-one alignment, keeping the points based on the minimum Haversine distance.

Watermark extraction. Watermark extraction is a process that retrieves the watermark from a watermarked trajectory. Broadly, a watermark extraction procedure can be blind and non-blind. Blind watermarking does not require the original data for watermark extraction, whereas non-blind watermarking requires the original data. Non-blind watermarking is typically more robust than blind watermarking and allows for larger watermarks [17]. The watermarking scheme considered in our work is non-blind.

The process of extracting the watermark is the reverse of the watermark embedding process. Given a watermarked trajectory $\widetilde{T}$, the extraction function $EXT(\cdot)$ extracts the watermark sequence $\widetilde{W}$ from the trajectory $\widetilde{T}$: $\widetilde{W} = EXT(T, \widetilde{T}, s)$. The watermark extraction function takes the original and the given trajectory as an input and outputs the extracted watermark sequence $\widetilde{W}$. More specifically, the trajectory $\widetilde{T}$ is partitioned into sub-trajectories, and then the FFT is applied to obtain the amplitude $\widetilde{\alpha}$. We retrieve the watermark by

$$\widetilde{w} = \frac{\widetilde{\alpha} - \alpha}{s}, \tag{5}$$

where $\widetilde{w} \in \widetilde{W}$, α is the amplitude of the candidate original trajectory T and s is the watermark strength.

Watermark correlation. The next step is to compute the correlation between the extracted watermark $\widetilde{w}$ and the original watermark w of each sub-trajectory to verify the watermark. To compute the correlation, we adopt Normalized Cross-Correlation (NCC), a commonly used measure for watermark verification [13, 21]. NCC's scale invariance properties ensure its effectiveness in comparing both the original watermark w and the extracted watermark $\widetilde{w}$. The NCC of two watermarks, w and $\widetilde{w}$, is defined by

$$\mathrm{NCC}(w, \widetilde{w}) = \frac{\sum_i w_i \widetilde{w}_i}{\sqrt{\sum_i w_i^2}\sqrt{\sum_i \widetilde{w}_i^2}}. \tag{6}$$

The value of NCC lies between -1 and 1. NCC value 1 indicates that two vectors are highly correlated, whereas 0 and -1 indicate no correlation and negative correlation, respectively. Finally, an average NCC score for all sub-trajectories of a given trajectory is calculated. The verification is successful if this score is higher than the acceptance threshold τ. We adopt $\tau > 0.85$ based on [32].

3.3 Computational Complexity Analysis

In this section, we discuss the computational complexity of the proposed watermark embedding and watermark verification algorithms.

3.3.1 Watermark Embedding. Our proposed watermark embedding algorithm presented in Section 3.1 splits the trajectory T into sub-trajectories with the complexity of $O(N/m)$, where N is the trajectory length and m is the sub-trajectory length. Next, *W-Trace* employs the FFT algorithm to embed watermarks into the sub-trajectories. For the trajectory T, FFT has $O(N \log_2 m)$ time complexity [6].

3.3.2 Watermark Verification. Our proposed watermark verification algorithm is presented in Section 3.2. In the first step, given the trajectory $\widetilde{T}$, we select a candidate trajectory T from the dataset using Haversine distance. The computational complexity of the selection and similarity computation is $O(d_n * N^2)$, where d_n is the number of trajectories in the dataset, and N is the average trajectory length. Next, we align the trajectory length T to $\widetilde{T}$ with a computational complexity of $O(N^2)$. For extracting the watermark, $\widetilde{T}$ is partitioned into sub-trajectories with $O(N/m)$ complexity. Then, FFT is applied to extract the watermark from $\widetilde{T}$ with a complexity of $O(N \log_2 m)$ [6]. Finally, the watermark correlation computation has a $O(N)$ complexity.

The quadratic complexity of candidate trajectory selection and trajectory length alignment can be further improved. For instance, deep learning-based methods such as t2vec [24] and STDRL [8] with a linear complexity can be utilized for the candidate trajectory search. Given a trained t2vec model, embedding a trajectory into a vector requires $O(N)$ time. Then, calculating the cosine similarity between the embedding vectors of two trajectories has a time complexity of $O(|v|)$, where $|v| << N$ is the length of the embedding vector. For trajectory length alignment, the nearest neighbor search based on spatial indexing can help to reduce the time complexity. The time complexity for constructing the spatial index using an R-tree is $O(N\ell ogN)$ [33]. Finding the nearest neighbor has a complexity of $O(N\ell ogN)$ [1].

4 THREAT MODEL: ATTACKS ON TRAJECTORIES

Digital watermarking is subject to adversarial modifications, denoted as attacks [34]. The objective of the adversarial modifications analyzed in this work is twofold: First, the adversary's goal is to prevent successful watermark verification, to obscure data origin and ownership. Second, the adversary who aims at illicit data monetization also seeks to retain the data utility for real-world applications. In general, adversary capabilities can include knowledge of the data, such as the

Algorithm 2 Watermark verification

Input: Original trajectory T, watermark sequence W, Trajectory $\widetilde{T}$, watermark strength s.
Output: $B \in \{true, false\}$.

1: Align the length of $\widetilde{T}$ to T using the Harvesine distance.
2: Split the trajectory $\widetilde{T}$ into fixed-length sub-trajectories $[\widetilde{t}_1, ..., \widetilde{t}_n]$.
3: **for** each $\widetilde{t} \in \widetilde{T}$ **do**
4: Represent the sub-trajectory $\widetilde{t}$ as a sequence of complex numbers, see Eq. (1).
5: Apply FFT and represent the result in polar coordinates $\widetilde{\alpha}, \widetilde{\varphi}$.
6: Extract watermark $\widetilde{w}$ from the sub-trajectory using the frequency representation of T, see Eq. (5).
7: Calculate the normalized cross-correlation (NCC) between extracted watermark $\widetilde{w}$ and original watermark w.
8: **end for**
9: Calculate the average watermark correlation of the trajectory $\widetilde{T}$
10: **return** true, if the average watermark correlation is greater than the threshold τ.

original and watermarked GPS trajectories, the watermark, and the watermarking algorithm. This knowledge can be further subdivided into perfect knowledge, where an adversary has access to all data, and limited knowledge, where an adversary has limited access [30]. In this work, we assume that an adversary has limited access, namely, knows the watermarked trajectory and the watermarking algorithm. In contrast, the original GPS data and the specific watermark embedded into the data remain unknown. This assumption is realistic in many real-world GPS trajectory watermarking applications. An adversary with limited knowledge cannot remove the watermark directly. Instead, an adversary can attempt heuristic trajectory modifications to prevent watermark verification. We refer to such modifications as attacks on trajectories.

The adversarial modifications of trajectory data are subject to a trade-off. Intuitively, when data undergoes substantial alterations, it can impede watermark verification, but this may also result in a reduction of the data's usefulness in practical applications, reducing data value. This trade-off is a constraint, restricting the extent to which the adversary can manipulate the data. To quantify the utility of the trajectory modified in the adversarial settings, we follow the same principle as we introduced for the trajectory watermarking and apply a modification threshold σ (see Definition 2.10):

$$\widetilde{T}' = AT(\widetilde{T}, \theta), \quad s.t.\ D(\widetilde{T}, \widetilde{T}') \leq \sigma.$$

Here, $AT(\cdot)$ is the attack function, $\widetilde{T}$ is the watermarked trajectory, θ represents the specific attack parameter, $D(\cdot)$ is the distance metric, $\widetilde{T}'$ is the modified watermarked trajectory, and σ is the modification threshold limiting the effects of the possible attacks on trajectories.

The attacks that we consider in this work have been discussed in the literature in the context of trajectory watermarking [10, 32], trajectory similarity measures [36], and cryptography [15]. In particular, we consider four different types of attacks: noise additive attacks, point replacement attacks, length modification attacks, and the combination of these types, the hybrid attack. As real-world trajectory data is noisy and comes with different sampling frequencies and lengths, we expect downstream applications to cope with the noise introduced by these attack types as long as the modification threshold is respected. A formal analysis of watermarking GPS trajectories is clearly out of the scope of our application-oriented approach, as GPS trajectories serve various downstream applications with distinct characteristics [11, 14]. Intuitively, even a strong or adaptive adversary without access to the original data and the watermark cannot directly optimize the

correlation or removal of the watermark. Instead, following [10, 32], we describe a reasonable set of attacks.

4.1 Noise Additive Attacks

Noise additive attacks involve adding random noise to the trajectory coordinates, as illustrated in Fig. 2a.

(1) **Additive Gaussian White Noise (AGWN).** In this attack, a value is drawn from a normal distribution randomly, which is then added to each GPS position in the trajectory.
(2) **Additive Signal to Noise Ratio (ASNR).** This attack shares similarities with AGWN, but the noise is scaled to achieve a specific signal-to-noise ratio (SNR) and added to each GPS position.
(3) **Additive Outliers with SNR (AOSNR).** We randomly select some points with a probability of $\theta = (p_{\text{AOSNR}})$ and apply the ASNR attack to these points.
(4) **Double Embedding Attack (DEA).** The double embedding attack aims to remove the original watermark from a watermarked trajectory by inserting a different watermark using the same method.

4.2 Point Replacement Attacks

To carry out point replacement attacks, certain elements of the trajectory are eliminated and replaced with information based on the adjacent points, as illustrated in Fig. 2b.

(1) **Replace Random Points (RRP).** The Replace Random Points attack involves selecting GPS coordinates with a probability of $\theta = (p_{\text{RRP}})$, and replacing them with the corresponding previous points.
(2) **Replace Random Points with Path (RRPP).** Similar to the Replace Random Points attack, each GPS coordinate is replaced with the probability $\theta = (p_{\text{RRPP}})$, and the replaced value is a convex combination of the remaining adjacent points.
(3) **Replace Non-Skeleton Points with Path (RNSPP).** The Ramer–Douglas–Peucker (RDP) algorithm [12] is utilized in this attack to identify the central points of the GPS trajectory that define its overall shape. The granularity of the remaining skeleton is controlled by a parameter $\theta = (\varepsilon)$. The removed points by the RDP algorithm are substituted by a convex combination of the neighboring points.

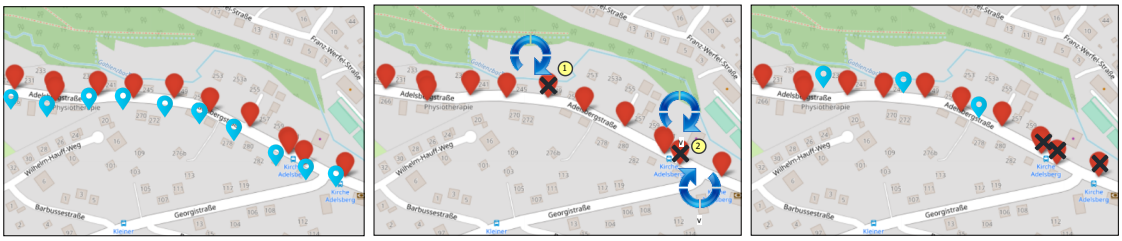
4.3 Length Modification Attacks

In contrast to previous attacks that maintain a constant trajectory length, length modification attacks involve altering the length of the trajectory by either cropping or interpolating the trajectory, as illustrated in Fig. 2c.

(1) **Linear Interpolation Attack (LIA).** The trajectory length is increased by inserting additional points at random positions through linear interpolation.
(2) **Cropping Attack (CA).** In a cropping attack, selected points are removed from the trajectory, decreasing the trajectory length.

4.4 Hybrid Attacks

Our experiments exemplify a hybrid attack as a combination of multiple attacks applied to the same trajectory to obscure the watermark. One example of a hybrid attack is a sequence of a cropping attack (CA) followed by additive Gaussian white noise (AGWN) and replace random points (RRP).



(a) (b) (c)

Fig. 2. (a) Noise additive attack is applied on the watermarked trajectory (red color), leading to the noisy trajectory generation (blue color). (b) Point replacement attacks. (1) A point is replaced by an adjacent point. (2) A point is replaced by a convex combination of the adjacent points. (c) A length modification attack is applied to the watermarked trajectory (red color). The cross mark shows the cropping attack, whereas new coordinates (blue color) are inserted with linear interpolation. Map data: ©OpenStreetMap contributors, ODbL.

5 EXPERIMENTAL SETUP

In this section, we describe the datasets, baselines, parameter settings, and evaluation metrics adopted in the evaluation.

5.1 Datasets

The proposed watermarking method is evaluated using two real-world anonymized trajectory datasets: the Porto and the German datasets. Moreover, we create a Candidate Trajectory Dataset to evaluate the effect of the candidate trajectory selection on the watermark verification. Furthermore, we generate a synthetic dataset to assess the utility of watermarking methods for downstream tasks that require user information, such as trajectory user linking.

Porto Dataset. The Porto dataset is publicly available and consists of trajectories of variable length generated by 442 taxis in the city of Porto, Portugal, from July 1, 2013, to June 30, 2014 [29]. We randomly sample 1,100 trajectories, each trajectory with a length of 256, from the Porto dataset, denoted as P. The trajectories have a sampling rate of four times per minute. The mean distance and the standard deviation between two consecutive GPS coordinates in the Porto dataset are 4.8 meters and 16 meters, respectively.

German Dataset. This dataset is provided by a proprietary data provider. The dataset comprises vehicle trajectory data from two German federal states, Saxony and Lower Saxony, with an average sampling rate of 12 times per minute. The data refers to September 2019. The average distance between two consecutive GPS coordinates is 68 meters, with a standard deviation of 56 meters. We randomly select 2,200 trajectories from the German dataset and divide them into two subsets: G_w with 1,100 trajectories to which watermarking is applied later in the experiment and G_{nw} with 1,100 non-watermarked trajectories.

Candidate Trajectory Dataset. The Candidate Trajectory Dataset G_c contains 2,200 trajectories and is divided into two subsets. The first subset, G_{cw}, consists of the trajectories from G_w, with each trajectory being watermarked and subjected to a randomly selected attack (as described in Section 4). The second subset, G_{nw}, contains 1,100 original trajectories from the German dataset without any modifications.

Synthetic Dataset. We utilize synthetic data to evaluate watermarking method performance in sensitive applications, such as trajectory user linking, without compromising user privacy. We generate synthetic GPS trajectories for New York City (NYC) over three months using the method proposed by Kim et al. [22]. The data contains time, latitude, longitude, and user ID. The dataset

includes approximately 400 synthetic users and their trajectories, segmented on a daily basis, denoted as S. We watermark the synthetic dataset and assess its utility in the trajectory user linking task.

5.2 Baselines

We compare the proposed *W-Trace* approach against two state-of-the-art watermarking methods from the audio domain and TrajGuard [32], a state-of-the-art method designed for GPS trajectories.
IMF Watermarking [13]. This watermarking technique is a non-blind method commonly used for watermarking audio signals. The trajectory data is transformed into a signal (latitude/longitude vs. time) and then decomposed into several components called Intrinsic Mode Functions (IMFs) using the Empirical Mode Decomposition (EMD) technique. Each IMF is a 1-D vector. Motivated by [13], the first IMF I is chosen and rewritten as a matrix of size $X \times Y$, where X is the number of rows and Y is the number of columns. This matrix is further decomposed into singular value matrices using Singular Value Decomposition (SVD). A watermark with a scaling factor d is then inserted into one of the matrices. An inverse EMD is applied to all watermarked IMF for watermarked trajectory generation. The verification steps are the reverse of the watermark embedding process.
TrajGuard [32]. TrajGuard is a blind watermarking method for GPS trajectories that utilizes geometric transformation. Initially, TrajGuard partitions the trajectory into multiple parts and then distributes the watermark into all the sub-trajectories. More specifically, TrajGuard performs spatial partition with size ω_s and temporal partition with interval ω_t for each sub-trajectory. Then, the centroid is calculated, and a watermarking vector is inserted with an intensity γ for each sub-trajectory by adjusting the distance of the points to the centroid.
SVD Watermarking [21]. Based on a blind audio watermarking approach, this method utilizes the SVD technique and the quantization index modulation method for both the insertion and verification of the watermark. First, the trajectory (2-D coordinates) is partitioned into non-overlapping 2-D fixed-size matrix blocks of size $U \times V$. Then, SVD is applied to each matrix block, which gives three singular matrices. For each block, Euclidean norms of singular values are computed. The watermark vector is embedded by quantizing the norm of each block using a quantization coefficient Δ. A reverse strategy is employed for the verification.

5.3 Parameter Settings

Here, we describe parameter settings for watermarking and attacks in the evaluation.

5.3.1 Method Parameters. We apply grid search to obtain optimal parameter settings for the watermarking methods considered in this work. The parameters are selected such that the average trajectory modification from watermarking remains below the modification threshold $\sigma = 10$ meters in all methods and datasets. The parameter selection details are described further in Appendix A.
W-Trace. We split each trajectory into sub-trajectories of the same length as the watermark. The watermark strength $s = 0.0003$ is obtained using grid search for all the datasets. We discuss the impact of the parameters in Section 6.4.
IMF watermarking. A watermark with similar length and values comparable to *W-Trace* is inserted into the trajectory. We perform a grid search on the IMF scaling factor d for all the datasets, resulting in $d = 0.00015$ for German (G) and Synthetic datasets (S) and $d = 0.0001$ for the Porto dataset (P).
TrajGuard. The parameters include the spatial partition ω_s and temporal partition ω_t to split the trajectory, as well as intensity γ. We follow Pan et al. [32] to select the TrajGuard parameter values based on the sampling rate. The resulting parameter values for the German (G_w) and Synthetic datasets (S) are $\omega_s = 0.03$, $\omega_t = 10$, and $\gamma = 0.0003$. For the Porto dataset (P), ω_t and γ values are the same as the German and Synthetic datasets, except ω_s with a value of 0.002.

SVD watermarking. A watermark comparable to *W-Trace* is inserted into the trajectory. We obtain $\Delta = 0.0003$ for all the datasets based on the grid search.

5.3.2 Attack Parameters. The parameter selection for the attacks described in Section 4 is based on the intuition that an adversary who intends to redistribute or monetize the data would not hamper the trajectory utility. Therefore, the parameters are set in a way that the distance between the watermarked trajectory $\widetilde{T}$ and the modified trajectory $\widetilde{T}'$ satisfies the modification threshold. We keep the attack parameters consistent in all the methods we evaluate. The variation of attack parameter values is discussed in Section 6.4.5.

(1) **AGWN.** For each trajectory T, we add a sample from a Gaussian distribution to each coordinate with mean $\mu = 0$ and variance $\sigma^2 = 0.00002$.
(2) **ASNR.** An SNR value of 105 is selected and inserted into each coordinate as discussed in Section 4.
(3) **AOSNR.** A specific number of data points are selected with probability $p_{\text{AOSNR}} = 0.03$ from each trajectory, and an ASNR attack is applied to those selected data points.
(4) **DEA.** A new random watermark is embedded into the trajectory with the *W-Trace* watermarking technique.
(5) **RRP.** In each trajectory, each point is replaced with their respective previous points with probability $p_{\text{RRP}} = 0.005$.
(6) **RRPP.** A set of points, selected with the probability $p_{\text{RRPP}} = 0.01$, are replaced by the average of the adjacent points.
(7) **RNSPP.** We use $\varepsilon = 10^{-6}$ for the RDP algorithm.
(8) **LIA.** We select three locations randomly from the trajectory. In each location, we insert a coordinate and timestamp generated by linear interpolation of the neighboring points.
(9) **CA.** Three data points with the last indices are removed.
(10) **Hybrid.** We utilize the same parameters as for the individual attacks.

5.4 Evaluation Metrics

To assess watermark verification effectiveness and robustness, we adopt watermark recognition rate, false-positive rate, average modification distance, and embedding capacity. These metrics are typically used to evaluate the performance of watermarking approaches. To assess the utility of watermarked trajectories, we adopt two downstream tasks, map matching and trajectory user linking. We utilize the Jaccard similarity coefficient and accuracy as evaluation metrics for map matching and trajectory user linking, respectively.

Watermark recognition rate. To evaluate the effectiveness and robustness of watermark verification, i.e., the capability to accurately identify a watermark in modified trajectory data, we utilize recognition rate. The recognition rate is the ratio of correctly identified watermarked trajectories (true positives or TP) to the total number of watermarked trajectories. The recognition rate is defined as follows: Recognition rate $= \frac{TP}{TP+FN}$, where FN is the number of false negatives, i.e., unrecognized watermarked trajectories. The recognition rate is also commonly referred to as recall and true positive rate. We report the recognition rate in %, where 100% corresponds to $FN = 0$.

To compute the recognition rate, we watermark all trajectories in each dataset. Then, we apply the adversarial modifications according to the threat model to all trajectories. We then use the corresponding watermark verification procedure to extract the watermark. We assess the similarity between the extracted and the embedded watermarks using the normalized cross-correlation (NCC), defined in Eq. 6. Following [32], we accept the watermark to be successfully verified if the average watermark correlation between the original watermark and extracted watermark from the noised trajectory is higher than the acceptance threshold, i.e., $\tau > 85\%$.

False-positive rate. The false-positive rate is defined as the ratio of trajectories without a watermark wrongly verified as watermarked to the total number of non-watermarked trajectories [20]: False-positive rate $= \frac{FP}{FP+TN}$. FP is the number of false positives, i.e., trajectories wrongly identified as watermarked, and TN is the number of true negatives, i.e., correctly recognized non-watermarked trajectories. We use the 1100 non-watermarked trajectories G_{nw} to compute the false-positive rate.

Average modification distance. We assess the trajectory modification distance resulting from watermarking methods as an average Haversine distance between the original and the modified trajectories.

Embedding capacity. We compute the watermark embedding capacity as the amount of watermark information embedded in the watermarked trajectory data. In this context, embedding capacity is defined as the ratio of the length of the watermark vector embedded and the total number of GPS points in the dataset.

Jaccard similarity coefficient. This coefficient calculates the similarity between two sets. In evaluating the map-matching task, we compute the similarity between the sets of matched street segments obtained using the original and the watermarked trajectories. The similarity score is calculated with the Jaccard similarity coefficient, where a score of 0 indicates no match, and a score of 1 indicates a perfect match.

Accuracy. The trajectory user linking (TUL) task utilizes accuracy as an evaluation metric. In this context, accuracy is defined as the proportion of trajectories correctly linked to their respective users, divided by the total number of trajectories in the dataset.

6 EVALUATION RESULTS

Our evaluation aims to assess the effectiveness and robustness of *W-Trace* under the threat model. Furthermore, we demonstrate the utility of the *W-Trace* watermarked trajectories for real-world applications. Then, we discuss the time complexity of the watermarking methods. We also analyze the impact of the method and attack parameters of *W-Trace* on the watermark recognition rate.

Table 2. Recognition rate of *W-Trace* and baseline methods on the Porto (P) and the German (G_w) datasets.

	Noise additive				Point replacement			Size mod.		Hybrid	Avg.
Method	AGWN	ASNR	AOSNR	DEA	RRP	RNSPP	RRPP	LIA	CA		
SVD_P	100.0	98.2	99.3	0.0	100.0	94.7	100.0	100.0	100.0	100.0	89.2
SVD_{G_w}	100.0	79.4	99.7	0.0	100.0	65.3	100.0	100.0	100.0	100.0	84.3
IMF_P	87.2	87.0	90.3	90.8	90.1	90.8	90.7	90.3	91.0	87.1	89.5
IMF_{G_w}	72.5	70.6	74.5	75.2	75.8	76.0	75.1	76.0	77.1	72.1	74.5
$TrajGuard_P$	59.8	56.2	55.7	61.7	68.3	65.0	68.3	63.6	64.5	57.5	62.1
$TrajGuard_{G_w}$	87.6	83.2	94.4	94.4	95.6	74.2	95.9	75.2	91.9	83.8	87.6
W-Trace$_P$	100.0	100.0	99.0	100.0	100.0	100.0	100.0	100.0	100.0	99.9	99.8
W-Trace$_{G_w}$	100.0	99.8	98.2	100.0	98.6	100.0	100.0	100.0	100.0	94.0	99.0

6.1 Effectiveness & Robustness

In this section, we evaluate the effectiveness and robustness of the proposed *W-Trace* approach. We compare our approach to the baseline methods on the German (G_w) and Porto datasets (P) under various attacks described in the threat model. We report various metrics, including recognition rate,

false-positive rate, and average modification distance. We compare the computational complexity of *W-Trace* with the baseline methods and discuss the amount of watermark information embedded into the GPS trajectories by different methods.

6.1.1 Watermark Recognition Rate. Table 2 demonstrates the effectiveness and robustness of our proposed *W-Trace* approach against all the considered attacks in the German (G_w) and Porto (P) datasets. The recognition rate of *W-Trace* in the German (*W-Trace*$_{G_w}$) and Porto datasets (*W-Trace*$_P$) averages around 99%, confirming the effectiveness, robustness, and generalizability of our approach.

Across the German and Porto datasets, the performance of baseline methods against some attacks varies. For instance, TrajGuard exhibits inconsistent performance against several attacks, particularly on the Porto dataset ($TrajGuard_P$). One reason why TrajGuard is more vulnerable to attacks on the Porto dataset is that this dataset has a higher spatial density than the German dataset, as mentioned in Section 5.1. This means there are more GPS points in a smaller area in the Porto dataset, making it more challenging for TrajGuard to detect and filter out the modifications introduced by attacks [32]. In addition, TrajGuard's vulnerability to attacks can be attributed to the fact that TrajGuard embeds a smaller amount of watermark information compared to *W-Trace*, which explains its lower recognition rate. The performance of the IMF watermarking method differed between the German and Porto datasets, with successful detection on the Porto dataset (IMF_P) but lower performance on the German dataset (IMF_{G_w}). The denser spatial area of the Porto dataset makes the decomposition process of the IMF method more effective, leading to an effective watermark verification process. In contrast, the larger geographical area covered by the German dataset makes it harder for the IMF method to decompose the trajectory into Intrinsic Mode Functions (IMFs) effectively, resulting in a lower recognition rate. We observe that the noise additive attack destroys the quantization-based watermark detection process in SVD watermarking. To summarize, unlike the baselines, the proposed *W-Trace* method is not impacted by the sparsity of the underlying data distribution and can effectively withstand the considered attacks.

6.1.2 False-Positive Rate. Our experiments on the German dataset (G_{nw}) demonstrate that the false-positive rate is 0% for the proposed *W-Trace* approach, as well as for the TrajGuard, and SVD baselines, demonstrating that these methods do not claim ownership for non-watermarked trajectories. In contrast, the IMF baseline claims 29.27% of non-watermarked trajectories as watermarked, making the IMF hardly applicable in real-world applications.

6.1.3 Average Modification Distance. The modification threshold defined in Eq. 2.10 bounds the distance between the original and the watermarked trajectory. We observe that in the configuration described in our evaluation design, all considered approaches respect the modification threshold and have a comparable average distance below 6 meters for German (G_w) and Porto (P) datasets, as illustrated in Fig. 3.

6.1.4 Embedding capacity. In this section, we analyze the amount of watermark information embedded in the watermarked trajectory data. The embedding capacity for the proposed *W-Trace* method, as well as for the IMF and SVD baseline methods, is 1, meaning that these methods embed one watermark value per GPS coordinate. For TrajGuard, the embedding capacity is approximately 0.25, meaning that, on average, one watermark value is inserted for every four GPS points. This suggests that TrajGuard embeds a lower amount of watermark information compared to the other considered watermarking approaches.

6.1.5 Summary. In summary, the proposed *W-Trace* approach is both effective and robust against all considered attacks, achieving a high recognition rate of 99% on average on the German (G_w) and Porto (P) datasets. In contrast, the baselines indicate varying performance and are more sensitive

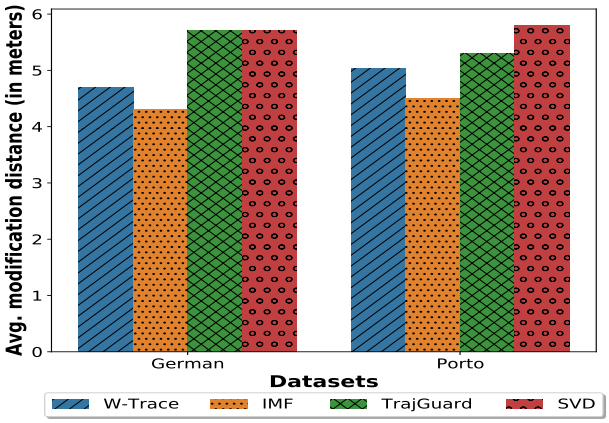


Fig. 3. Average modification distance between the original and the watermarked trajectories on the German (G_w) and Porto (P) datasets.

to factors such as dataset density and attacks. Compared to the TrajGuard, *W-Trace* embeds more watermark information into the trajectory, leading to higher robustness. More specifically, *W-Trace* embeds four times more watermark information than TrajGuard. Furthermore, *W-Trace* indicates a zero false-positive rate in our evaluation.

6.2 Utility: Effect of Watermarking on Real-World Applications

As outlined in Section 2, the proposed watermarking approach is designed to preserve the data utility for real-world applications. Consequently, downstream applications should be able to utilize the data despite the modifications introduced by watermarking effectively. In this section, we evaluate the impact of watermarking on real-world applications, including map matching and trajectory user linking.

6.2.1 Map matching. We select map matching to evaluate trajectory utility, given its significance as the initial processing step in numerous mobility applications, such as traffic speed prediction and accident prediction. To evaluate the impact of the watermarking methods on map matching, we filter trajectories from the German dataset (G_w) for a specific geographic region, namely, the federal state of Saxony described in Section 5.1, resulting in 767 GPS trajectories. We watermark the resulting trajectories using our proposed *W-Trace* approach and baseline methods. Then, we apply a state-of-the-art map matching algorithm[2] to the original and the watermarked trajectories resulting from the watermarking methods. The trajectories watermarked with *W-Trace* exhibit a high average similarity score of 0.98, which confirms that watermarked trajectories resulting from our approach match approximately the same street segments as the street segments of the original trajectories, as depicted in Table 3. The TrajGuard approach attains an average similarity score of approximately 0.99. The high Jaccard coefficient score of TrajGuard is due to the inclusion of 75% GPS coordinates in the watermarked trajectories from the original trajectories, as mentioned in Section 6.1.4. In contrast, the average Jaccard coefficient value for IMF and SVD methods falls below 0.90, indicating that watermarked trajectories resulting from these methods have lower utility regarding map matching. In summary, our proposed *W-Trace* approach preserves data utility for map-matching applications while ensuring robustness and embedding a significant amount of watermark information compared to the baseline methods.

[2]https://github.com/valhalla/valhalla

Table 3. Utility of *W-Trace* and baseline methods for modified trajectories.

Watermarking method	Applications		
	Map Matching (Jaccard Similarity Coefficient)	TUL (watermarked) (Accuracy)	TUL (attacked) (Accuracy)
SVD	0.739 ± 0.139	0.982 ± 0.020	0.970 ± 0.020
IMF	0.894 ± 0.100	0.973 ± 0.017	0.970 ± 0.019
TrajGuard	0.990 ± 0.001	0.973 ± 0.018	0.971 ± 0.023
W-Trace	0.980 ± 0.003	0.979 ± 0.019	0.972 ± 0.022

6.2.2 Trajectory user linking (TUL). The identification of users based on their mobility behavior is crucial for various applications, such as point of interest (POI) recommendation, predicting the next location, and monitoring the COVID-19 pandemic [8]. We aim to assess how watermarking impacts a TUL model. Furthermore, we aim to assess the influence of attacks on the utility of watermarked GPS trajectories for TUL.

To demonstrate the utility of watermarked trajectories in the TUL context, we employ the state-of-the-art trajectory user linking method, TULAM [23]. TULAM is a neural network-based approach with an attention-based mechanism that leverages historical GPS trajectories. TULAM incorporates GPS coordinates as a feature in the model architecture, which enables us to verify the impact of watermarking on the data utility for the TULAM model. To enable our evaluation, we watermark the NYC synthetic trajectory dataset (S) with our proposed *W-Trace* approach and the baseline methods. Then, we apply the hybrid attack presented in Section 4 to the watermarked trajectories.

We train the TULAM model on the original, watermarked, and attacked trajectories. We apply k-fold cross-validation with k=30, and compute the average accuracy score. The results are illustrated in Table 3. The TULAM model trained on original synthetic trajectory data achieves an average accuracy score of approximately 0.974. This high accuracy is attributed to the widespread distribution of trajectories across a large region (NYC), where users exhibit distinct mobility patterns, facilitating effective user identification. In comparison, the TULAM model trained on watermarked trajectories resulting from our proposed *W-Trace* approach attains an average accuracy score for user linking of around 0.979. To further analyze the statistical significance of the accuracy score, we perform a paired t-test between the accuracy scores of the original and watermarked trajectories. The outcomes of paired t-tests for our *W-Trace* approach reveal that these differences are statistically insignificant (p-value > 0.05), confirming that watermarked trajectories from *W-Trace* approach preserve the utility of the watermarked trajectory. We observe similar paired t-test results for watermarking trajectories resulting from the baseline methods. Next, we apply the hybrid attack on the watermarked trajectories resulting from different methods and assess the utility of attacked trajectories. The result demonstrates that the attacked trajectories also retain the utility for the trajectory user linking tasks. In conclusion, our *W-Trace* approach effectively preserves the utility for both watermarked and attacked trajectories for the TUL task.

6.3 Analysis of the Watermarking Time Complexity

In this section, we perform a comparative analysis of the time complexity of watermarking methods. Our proposed watermarking algorithm utilizes the FFT algorithm to embed watermarks in the GPS trajectory. FFT has $O(N \log_2 m)$ [6] time complexity, as discussed in Section 3.3. In the SVD watermarking method, the trajectory (2-D coordinates) is partitioned into non-overlapping 2-D

matrix blocks of size $U \times V$. The time complexity of the SVD method is $O(UVmin(U,V))$ [38]. For IMF watermarking, each trajectory is first decomposed with Empirical mode decomposition (EMD) into IMF. The time complexity of the EMD algorithm is similar to the FFT algorithm, i.e., $O(N\ell og_2 N)$ [40]. Further, an IMF is represented as a matrix $X \times Y$ and decomposed with the SVD algorithm, which has time complexity $O(XYmin(X,Y))$. So, the overall time complexity of the IMF method is $O((N \log_2 N) + O(XYmin(X,Y))$. Our proposed *W-Trace* watermarking method has lower time complexity than the IMF and SVD methods. The TrajGuard watermarking approach has a complexity $O(N)$ [32]. However, there is a trade-off between time complexity and robustness against the attacks. The linear time complexity of TrajGuard comes at the price of lower robustness compared to our proposed approach.

6.4 Parameter Analysis

In our evaluation, we assess the impact of different parameters on the effectiveness and robustness of our approach.

6.4.1 Effect of Watermark Strength. We evaluate *W-Trace* with different watermark strength s, defined in Eq. (3). We experiment with different values of s, i.e., 0.0001, 0.0003 and 0.0009, and assess the recognition rate of *W-Trace* as illustrated in Fig. 4. We keep the parameters of the attacks described in Section 5.3 consistent in all the settings. At a low watermark strength of $s = 0.0001$, the recognition rate is also low, with only 0-27% for several attacks. As expected, the recognition rate increases with the increasing s [37]. At $s = 0.0003$ and $s = 0.0009$, the minimum recognition rate across the attacks is 94% and 99.91%, respectively.

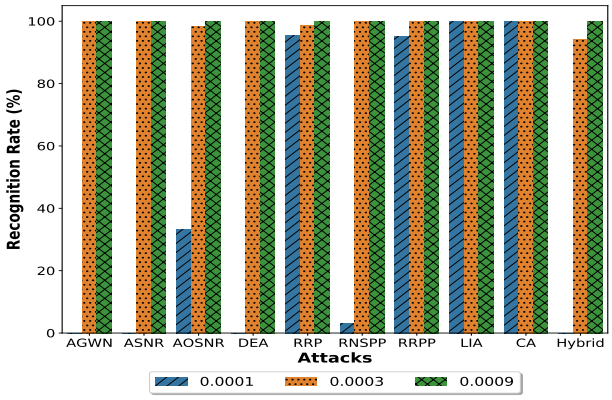


Fig. 4. Effect of watermark strength on the *W-Trace* recognition rate under different attacks on the German dataset (G_w).

6.4.2 Effect of Watermark Dimensionality. *W-Trace* inserts the watermark of specific dimensionality into the sub-trajectories of each trajectory. To assess the effect of watermark dimensionality on the *W-Trace* recognition rate, we insert watermarks with dimensionalities $m = \{8, 16, 32\}$ into sub-trajectories. When the watermark dimensionality is low, i.e., eight, *W-Trace* performs well under RRPP and length modification attacks. However, with such a short watermark length, *W-Trace* cannot verify the watermarks under other attacks. Fourier's descriptors of short watermarked sub-trajectories are sensitive to external noise, causing the sub-trajectory to shift from the original position beyond the modification threshold (σ). As the watermark dimensionality increases to 16 and 32, the average modification remains below the threshold, and the *W-Trace* approach resists all the attacks, as illustrated in Fig. 5.

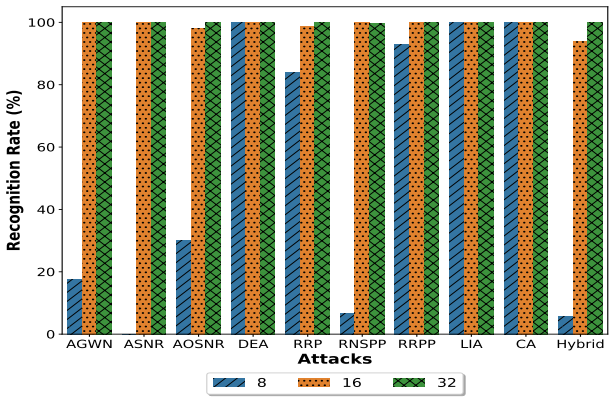
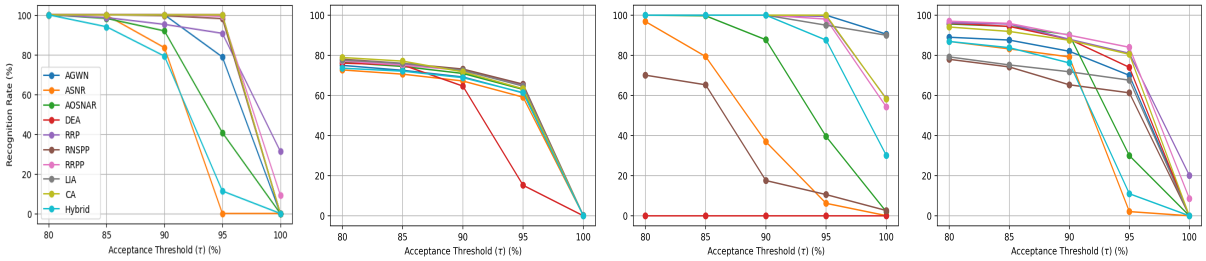


Fig. 5. Effect of watermark length on the *W-Trace* recognition rate under different attacks on the German dataset (G_w).

6.4.3 Effect of Acceptance Threshold. We evaluate the effect of varying acceptance thresholds, as defined in Section 5.4, on the recognition rate of different approaches using the German dataset (G_w). We vary the acceptance threshold value as $\tau = \{> 80\%, > 85\%, > 90\%, > 95\%, 100\%\}$, as illustrated in Fig. 6. At lower acceptance thresholds ($> 80\%$), all methods consistently achieve high recognition rates for different attacks. However, all methods experience a significant decline in recognition rates at thresholds of 95% and above. In particular, the IMF method demonstrates lower recognition rates compared to the other approaches at different τ values. In our experiments, we set the acceptance threshold value (τ) to 85%.



(a) *W-Trace* (b) IMF Watermarking (c) SVD Watermarking (d) TrajGuard

Fig. 6. Effect of varying acceptance threshold on recognition rate under various attacks on the German dataset (G_w).

6.4.4 Effect of the candidate trajectory selection on watermark verification. In this section, we evaluate the effect of the candidate trajectory selection step on the watermark verification, described in Section 3.2. In particular, we aim to determine the impact of the candidate selection based on the Haversine distance on the watermark recognition rate and the false-positive rate. In practice, the candidate trajectory selection step may fail to identify the matching original trajectory in two scenarios: 1) another trajectory in the dataset has a smaller Haversine distance. This issue can be mitigated by selecting more trajectory candidates (e.g., top-k), and 2) the trajectory of interest does not exist in the dataset, such that the selection based on a distance metric returns a non-matching trajectory. Given a non-matching trajectory pair provided for the watermark verification, the intended behavior of the verification procedure is to fail to verify the watermark, i.e., to have a low false-positive rate. We compare the performance of our proposed *W-Trace* approach with the IMF watermarking, a non-blind baseline method.

Table 4. The watermark verification results for IMF and *W-Trace* on the candidate trajectory dataset G_c. The table presents the total number of trajectories in G_c, the percentage of trajectories in G_{cw} correctly matched with G_w, the recognition rate, and the false-positive (FP) rate.

Method	# Traj., G_c	Traj. in G_{cw} correctly matched with G_w, %	Recognition rate, %	FP rate, %
IMF	2,200	100	71.27	29.27
W-Trace	2,200	100	99.00	0.00

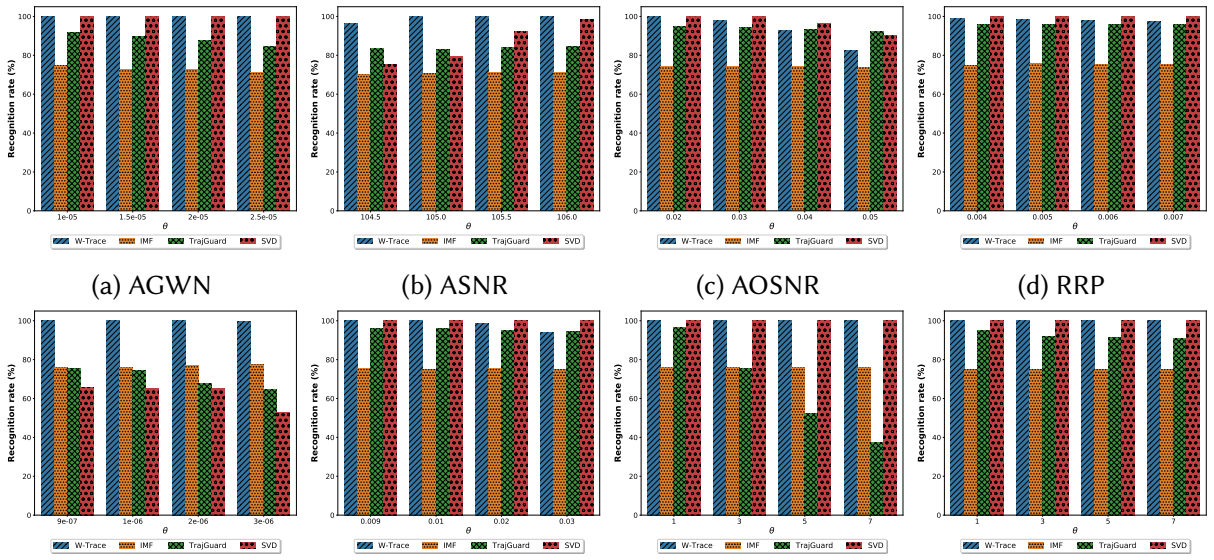
In this experiment, we utilize two datasets: the German dataset G_w and the candidate trajectory dataset G_c, described in Section 5.1. The German dataset G_w contains original trajectories. In these settings, the trajectory owner maintains this dataset for watermark verification purposes. The dataset G_c contains watermarked and attacked trajectories G_{cw} derived from G_w and non-watermarked trajectories G_{nw}. For each trajectory in $\widetilde{T} \in G_c$, the candidate selection step aims to identify the closest matching trajectory from the German dataset $T \in G_w$ using the minimum Haversine distance, as discussed in Section 3.2. The results are presented in Table 4. Our results indicate that all the trajectories in G_{cw} correctly match their true pairs $T \in G_w$ in both approaches. In this setting, the selected candidate pairs for G_{nw} will not match by design. We extract the watermark from trajectories in G_c based on the identified pairs and perform the watermark extraction step. As expected, the results in Table 4 demonstrate a 99% watermark recognition rate for the *W-Trace* approach and 71.27% for IMF watermarking, which are consistent with the results in Table 2. Moreover, our approach successfully identifies non-watermarked trajectories with a zero false-positive rate, while IMF watermarking exhibits a 29.27% false-positive rate. These results demonstrate that our *W-Trace* approach is effective and robust in distinguishing between watermarked and non-watermarked trajectories and outperforms the IMF watermarking by a large margin.

6.4.5 Variation of Attack Parameters. We evaluate the recognition rate of different methods with varying attack parameters (θ) on the German dataset (G_w). Furthermore, we vary the number of inserted and removed data points in LIA and CA attacks, respectively. The *W-Trace* method is robust to all the attacks with different values of the parameter, as illustrated in Fig. 7. The IMF watermarking method does not perform well in all the attacks. In addition, TrajGuard and SVD methods have the lowest recognition rate in RNSPP attacks.

7 RELATED WORK

This section discusses state-of-the-art methods of watermarking from the media and mobility domains. While most current watermarking research focuses on the media domain, including audio, video, and images, watermarking in the mobility domain, and, in particular, watermarking GPS trajectories, remains relatively limited. Furthermore, we briefly discuss representation learning methods for GPS trajectories.

Watermarking in the media domain. In the audio domain, El-Wahab et al. [13] employed Empirical Mode Decomposition (EMD) to decompose the signal into multiple Intrinsic Mode Functions (IMFs) and added the watermark vector to one of the IMFs using Singular Value Decomposition (SVD). Similarly, K. et al. [20] developed a blind adaptive audio watermarking algorithm based on SVD and utilized the Discrete Wavelet Transform (DWT). DWT and Discrete Cosine Transform (DCT) techniques are used in watermarking in the image domain [3, 5]. Additionally, deep learning-based approaches have been utilized for encoding the watermark in images, such as adversarial training and channel coding [27]. In the video domain, watermarking applications are explored,



(a) AGWN (b) ASNR (c) AOSNR (d) RRP

(e) RNSPP (f) RRPP (g) LIA (h) CA

Fig. 7. Recognition rate of different methods with varying attacking parameter (θ) on the German dataset (G_w).

such as the VStegNET method [28], which extracts spatio-temporal features using 3D-CNN to embed watermark information. Luo et al. [26] proposed an adversarial training based end-to-end trainable framework called DVMark for video watermarking. In this article, we utilize methods from the audio domain [13, 21] as baselines. Experimental results reveal that our proposed approach is more robust than audio domain-based methods.

Watermarking GPS trajectories. The research on watermarking GPS trajectories is still limited. Jin et al. [19] introduced an initial watermarking technique that involves inserting a small error into the GPS coordinates that define the trajectory shape. This technique has limitations, such as its inability to function effectively when consecutive similar coordinates exist, such as stops within trajectory data. In contrast, the state-of-the-art TrajGuard approach for watermarking GPS trajectories by Pan et al. [32] utilizes a geometric transformation to watermark GPS trajectories by partitioning trajectories into sub-trajectories and embedding the watermark using the centroid distance. In this article, we utilize TrajGuard as a baseline. We demonstrate through experiments that our proposed approach, *W-Trace*, is more robust, effective, utility-preserving, and capable of embedding more information.

Representation learning for GPS trajectories. Representation learning methods such as [24], [42], [35] create low-dimensional vector representations of GPS trajectories. These methods embed trajectories into low-dimensional vector spaces to preserve trajectory properties essential for similarity computation. Such methods can assess trajectory similarity in the presence of noise [24] or adversarial attacks [35]. However, unlike watermarking, such methods do not embed explicit provenance information into the trajectories. Therefore, trajectory similarity computation based on latent representations is less interpretable for provenance assessment than watermark verification.

8 CONCLUSION

In this article, we propose *W-Trace* – a novel, effective, robust, and utility-preserving approach for watermarking GPS trajectories. *W-Trace* adopts a Fourier-based technique and embeds the watermark using the Discrete Fourier Transform in the complex domain. *W-Trace* achieves an

average watermark recognition rate of around 99%. Our results indicate that the watermarked trajectories generated by the *W-Trace* approach retain relevant utility characteristics of the original trajectories. Moreover, *W-Trace* embeds more watermark information than the baseline methods.

ACKNOWLEDGMENTS

This work was partially funded by the German Research Foundation under "WorldKG" (424985896), the European Commission (EU H2020) under "smashHit" (871477), the B-IT foundation, and the state of North Rhine-Westphalia (Germany).

REFERENCES

[1] Elke Achtert, Christian Böhm, Peer Kröger, Peter Kunath, Alexey Pryakhin, and Matthias Renz. 2006. Efficient reverse k-nearest neighbor search in arbitrary metric spaces. In *Proceedings of the ACM SIGMOD International Conference on Management of Data 2006*. ACM, 515–526.

[2] Ramin Almasi and Hooman Nikmehr. 2015. A High-Payload Audio Watermarking Scheme for Real-Time Applications. *Journal of Computing and Security* (2015), 119–127.

[3] Ali Benoraira, Khier Benmahammed, and Noureddine Boucenna. 2015. Blind image watermarking technique based on differential embedding in DWT and DCT domains. *EURASIP J. Adv. Signal Process.* (2015), 1–11.

[4] David M Bevly. 2004. Global positioning system (GPS): A low-cost velocity sensor for correcting inertial sensor errors on ground vehicles. *Journal of dynamic systems, measurement, and control* (2004), 255–264.

[5] Uzair Aslam Bhatti, Linwang Yuan, Zhaoyuan Yu, Jingbing Li, Saqib Ali Nawaz, Anum Mehmood, and Kun Zhang. 2021. New watermarking algorithm utilizing quaternion Fourier transform with advanced scrambling and secure encryption. *Multim. Tools Appl.* (2021), 13367–13387.

[6] Leo I. Bluestein. 1970. A linear filtering approach to the computation of discrete Fourier transform. *IEEE Transactions on Audio and Electroacoustics* (1970), 451–455.

[7] Manuel Cedillo-Hernandez, Antonio Cedillo-Hernandez, and Francisco J Garcia-Ugalde. 2021. Improving dft-based image watermarking using particle swarm optimization algorithm. *Mathematics* (2021).

[8] Wei Chen, Shuzhe Li, Chao Huang, Yanwei Yu, Yongguo Jiang, and Junyu Dong. 2022. Mutual Distillation Learning Network for Trajectory-User Linking. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022*. ijcai.org, 1973–1979.

[9] Rajjat Dadwal, Thorben Funke, and Elena Demidova. 2021. An Adaptive Clustering Approach for Accident Prediction. In *Proceedings of the 24th IEEE International Intelligent Transportation Systems Conference, ITSC 2021*. IEEE, 1405–1411.

[10] Rajjat Dadwal, Thorben Funke, Michael Nüsken, and Elena Demidova. 2022. W-trace: robust and effective watermarking for GPS trajectories. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2022*. ACM, 77:1–77:4.

[11] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. 2013. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports* (2013), 1–5.

[12] David H. Douglas and Thomas K. Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10 (1973), 112–122.

[13] Basant S. Abd El-Wahab, Heba Ali El-Khobby, Mustafa M. Abd-Elnaby, and Fathi E. Abd El-Samie. 2021. Simultaneous speaker identification and watermarking. *International Journal of Speech Technology* (2021), 205–218.

[14] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2010. Show me how you move and I will tell you who you are. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS, SPRINGL 2010*. ACM, 34–41.

[15] Raju Halder, Shantanu Pal, and Agostino Cortesi. 2010. Watermarking Techniques for Relational Databases: Survey, Classification and Comparison. *J. Univers. Comput. Sci.* (2010), 3164–3190.

[16] Bing He, Dian Zhang, Siyuan Liu, Hao Liu, Dawei Han, and Lionel M. Ni. 2018. Profiling Driver Behavior for Personalized Insurance Pricing and Maximal Profit. In *Proceedings of the IEEE International Conference on Big Data (IEEE BigData 2018)*. IEEE, 1387–1396.

[17] Amir Houmansadr, Negar Kiyavash, and Nikita Borisov. 2009. RAINBOW: A Robust And Invisible Non-Blind Watermark for Network Flows. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2009*. The Internet Society.

[18] Xiaowei Hu, Shi An, and Jian Wang. 2018. Taxi driver's operation behavior and passengers' demand analysis based on GPS data. *Journal of advanced transportation* (01 2018), 1–11.

[19] Xiaoming Jin, Zhihao Zhang, Jianmin Wang, and Deyi Li. 2005. Watermarking Spatial Trajectory Database. In *Proceedings of the Database Systems for Advanced Applications, 10th International Conference, DASFAA 2005*. Springer, 56–67.

[20] Vivekananda Bhat K., Indranil Sengupta, and Abhijit Das. 2010. An adaptive audio watermarking based on the singular value decomposition in the wavelet domain. *Digit. Signal Process.* (2010), 1547–1558.

[21] Vivekananda Bhat K., Indranil Sengupta, and Abhijit Das. 2011. A New Audio Watermarking Scheme Based on Singular Value Decomposition and Quantization. *Circuits Syst. Signal Process.* (2011), 915–927.

[22] Joon-Seok Kim, Hyunjee Jin, Hamdi Kavak, Ovi Chris Rouly, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. 2020. Location-based social network data generation based on patterns of life. In *Proceedings of the 2020 21st IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 158–167.

[23] Hao Li, Shuyu Cao, Yaqing Chen, Min Zhang, and Dengguo Feng. 2024. TULAM: trajectory-user linking via attention mechanism. *Science China Information Sciences* (2024), 1–18.

[24] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S. Jensen, and Wei Wei. 2018. Deep Representation Learning for Trajectory Similarity Computation. In *Proceedings of the 34th IEEE International Conference on Data Engineering, ICDE 2018*. IEEE Computer Society, 617–628.

[25] Zhishuo Liu, Qianhui Shen, Han Li, and Jingmiao Ma. 2017. A Risky Driving Behavior Scoring Model for the Personalized Automobile Insurance Pricing. In *Proceedings of the 2nd International Conference on Crowd Science and Engineering, ICCSE 2017*. ACM, 61–67.

[26] Xiyang Luo, Yinxiao Li, Huiwen Chang, Ce Liu, Peyman Milanfar, and Feng Yang. 2021. DVMark: A Deep Multiscale Framework for Video Watermarking. *CoRR* (2021). arXiv:2104.12734

[27] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. 2020. Distortion Agnostic Deep Watermarking. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*. Computer Vision Foundation / IEEE, 13545–13554.

[28] Aayush Mishra, Suraj Kumar, Aditya Nigam, and Saiful Islam. 2019. VStegNET: Video Steganography Network using Spatio-Temporal features and Micro-Bottleneck. In *Proceedings of the 30th British Machine Vision Conference 2019, BMVC 2019*. BMVA Press, 274.

[29] Luís Moreira-Matias, Michel Ferreira, and João Mendes Moreira. 2015. Taxi Service Trajectory - Prediction Challenge, ECML PKDD 2015.

[30] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. 2017. Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017*. ACM, 27–38.

[31] Henri J Nussbaumer. 1981. The fast Fourier transform. In *Fast Fourier Transform and Convolution Algorithms*. Springer, 80–111.

[32] Zheyi Pan, Jie Bao, Weinan Zhang, Yong Yu, and Yu Zheng. 2019. TrajGuard: A Comprehensive Trajectory Copyright Protection Scheme. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019*. ACM, 3060–3070.

[33] Jianzhong Qi, Yufei Tao, Yanchuan Chang, and Rui Zhang. 2018. Theoretically Optimal and Empirically Efficient R-trees with Strong Parallelizability. *Proc. VLDB Endow.* 11, 5 (2018), 621–634.

[34] Erwin Quiring, Daniel Arp, and Konrad Rieck. 2018. Forgotten Siblings: Unifying Attacks on Machine Learning and Digital Watermarking. In *Proceedings of the 2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018*. IEEE, 488–502.

[35] Stefan Schestakov, Simon Gottschalk, Thorben Funke, and Elena Demidova. 2024. RE-Trace: Re-Identification of Modified GPS Trajectories. *ACM Trans. Spatial Algorithms Syst.* (feb 2024).

[36] Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. 2020. A survey of trajectory distance measures and performance evaluation. *VLDB J.* (2020), 3–32.

[37] Hai Tao, Li Chongmin, Jasni Mohamad Zain, and Ahmed N Abdalla. 2014. Robust image watermarking theories and techniques: A review. *Journal of applied research and technology* (2014), 122–138.

[38] Vinita Vasudevan and M. Ramakrishna. 2017. A Hierarchical Singular Value Decomposition Algorithm for Low Rank Matrices. *CoRR* (2017). arXiv:1710.02812

[39] Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*. ACM, 25–34.

[40] Yung-Hung Wang, Chien-Hung Yeh, Hsu-Wen Vincent Young, Kun Hu, and Men-Tzung Lo. 2014. On the computational complexity of the empirical mode decomposition algorithm. *Physica A: Statistical Mechanics and its Applications* (2014), 159–167.

[41] Shmuel Winograd. 1978. On computing the discrete Fourier transform. *Mathematics of computation* (1978), 175–199.

[42] Di Yao, Haonan Hu, Lun Du, Gao Cong, Shi Han, and Jingping Bi. 2022. TrajGAT: A Graph-based Long-term Dependency Modeling Approach for Trajectory Similarity Computation. In *Proceedings of the 28th ACM SIGKDD Conference on*

Knowledge Discovery and Data Mining, KDD 2022. ACM, 2275–2285.

[43] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. 2018. HiDDeN: Hiding Data With Deep Networks. In *Proceedings of the Computer Vision - ECCV 2018 - 15th European Conference (Lecture Notes in Computer Science)*. Springer, 682–697.

A PARAMETER SELECTION

In this section, we describe a grid search to select the watermarking method parameters for three datasets. In particular, we perform a grid search over watermark strength s in *W-Trace*, scaling factor d in IMF, quantization coefficient Δ in SVD, and intensity γ in TrajGuard, such that the average modification distance between the original and watermarked trajectories is below the modification threshold and comparable across the watermarking methods. We use Haversine distance as a distance measure. Table 5 presents the average modification distance for different methods with different parameter values for the German (G_w), Porto (P), and Synthetic (S) datasets. The selected parameters are mentioned in bold in the table.

Table 5. Average modification distance (in meters) between the original trajectory and watermarked trajectory for the German (G_w), Porto (P), and Synthetic (S) dataset

Watermarking Method	Parameter	German (G)	Porto (P)	Synthetic (S)
W-Trace	$s = 0.00025$	3.9	4.2	3.1
	$\mathbf{s = 0.0003}$	**4.7**	**5.0**	**5.0**
	$s = 0.00035$	5.5	5.9	6.0
	$s = 0.0004$	6.3	6.7	**6.3**
IMF	$d = 0.0001$	2.4	**4.5**	3.0
	$\mathbf{d = 0.00015}$	**4.3**	6.9	**5.2**
	$d = 0.0002$	6.4	9.5	7.2
	$d = 0.00025$	8.7	12.0	9.1
TrajGuard	$w_s = 0.002$(P), $w_s = 0.03$(G_w & S), $w_t = 10, \gamma = 0.0001$	1.7	1.7	1.8
	$w_s = 0.002$(P), $w_s = 0.03$(G_w & S), $w_t = 10, \gamma = 0.0002$	3.4	3.5	3.0
	$\mathbf{w_s = 0.002}$(P), $\mathbf{w_s = 0.03}$(G_w & S), $\mathbf{w_t = 10}, \gamma = \mathbf{0.0003}$	**5.7**	**5.3**	**5.0**
	$w_s = 0.002$(P), $w_s = 0.03$(G_w & S), $w_t = 10, \gamma = 0.0004$	6.9	7.0	7.1
SVD	$\Delta = 0.0002$	3.8	3.9	2.9
	$\mathbf{\Delta = 0.0003}$	**5.7**	**5.8**	**5.4**
	$\Delta = 0.0004$	7.7	7.8	7.1
	$\Delta = 0.0005$	9.6	9.6	9.0

Appendix D

Publication: W-Trace: Robust and Effective Watermarking for GPS Trajectories

Rajjat Dadwal, Thorben Funke, Michael Nüsken, and Elena Demidova

The 30th International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2022

DOI: 10.1145/3557915.3561474

Rajjat Dadwal, Thorben Funke, Michael Nüsken, and Elena Demidova, "W-Trace: Robust and Effective Watermarking for GPS Trajectories", The 30th International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2022. The original version of the record can be found at: https://dl.acm.org/doi/10.1145/3557915.3561474.

W-Trace: Robust and Effective Watermarking for GPS Trajectories

Rajjat Dadwal[1], Thorben Funke[1], Michael Nüsken[2], Elena Demidova[3]
[1]L3S Research Center, Leibniz University Hannover, Hannover, Germany
[2]Bonn-Aachen International Center for Information Technology, Bonn, Germany
[3]Data Science and Intelligent Systems Group (DSIS), University of Bonn, Bonn, Germany
dadwal@L3S.de,tfunke@L3S.de,nuesken@bit.uni-bonn.de,elena.demidova@cs.uni-bonn.de

ABSTRACT

With the rise of data-driven methods for traffic forecasting, accident prediction, and profiling driving behavior, personal GPS trajectory data has become an essential asset for businesses and emerging data markets. However, as personal data, GPS trajectories require protection. Especially by data breaches, verification of GPS data ownership is a challenging problem. Watermarking facilitates data ownership verification by encoding provenance information into the data. GPS trajectory watermarking is particularly challenging due to the spatio-temporal data properties and easiness of data modification; as a result, existing methods embed only minimal provenance information and lack robustness. In this paper, we propose *W-Trace* – a novel GPS trajectory watermarking method based on Fourier transformation. We demonstrate the effectiveness and robustness of *W-Trace* on two real-world GPS trajectory datasets.

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; • **Security and privacy**;

KEYWORDS

GPS trajectory, Watermarking, Data provenance, Data protection

1 INTRODUCTION

Personal GPS trajectory data are adopted in various critical domains, including data-driven urban traffic management, mobility, communication, and health. However, GPS trajectory data encode sensitive personal information such as user addresses, visited locations, and routes. Sharing and trading personal GPS trajectory data, even based on user consent, can occasionally result in data breaches and user privacy loss [3].

Figure 1 illustrates an example application scenario in which GPS trajectory data, initially shared according to the user's consent, is obtained by an adversary due to a data breach, modified to obscure the data origin, and illegally re-distributed on the market. Whereas the modification makes it challenging to claim the data ownership and to identify the misuse, sensitive personal information, such as user routes and driving patterns encoded in the trajectory, remains visible. The risk of data breaches necessitates the development of effective and robust provenance information

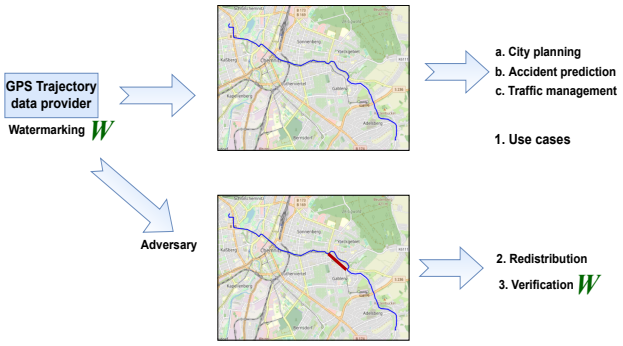


Figure 1: An example *W-Trace* application scenario. Watermarked GPS trajectory data is modified and re-distributed by an adversary. *W-Trace* enables data provenance verification. Map data: ©OpenStreetMap contributors, ODbL.

embedding methods for personal GPS trajectory data to facilitate data provenance verification.

Digital watermarking refers to methods that embed provenance information (so-called watermarks) into noise-tolerant data. Watermarking has been extensively studied in the media domain to protect images, videos, and audio files [2, 4]. In contrast, only a few initial approaches target watermarking of personal GPS trajectories [6, 9]. Watermarking GPS trajectories poses several challenges and is an inherently difficult task. The strength of a watermark is subject to a trade-off. On the one hand, a watermark should be robust, i.e., strong enough not to be removed by an adversary. On the other hand, a watermark should, at the same time, be weak, such that the watermarked data is still usable in the downstream applications. In addition to this general challenge for digital watermarking, GPS trajectories are, with their non-uniform sampling rate and positional inaccuracy, inherently susceptible to different modifications than media data, such as removal/addition of points or re-sampling along the path. State-of-the-art watermarking methods in the trajectory domain either lack robustness [6] or are ineffective, i.e., they embed only a small amount of data [9].

In this paper, we propose *W-Trace* – a novel, robust and effective watermarking method for personal GPS trajectories. *W-Trace* represents two-dimensional trajectory coordinates as complex numbers and adopts Discrete Fourier Transform (DFT) to enable effective watermark embedding in the frequency domain. To the best of our knowledge, we are the first to propose a DFT-based watermarking scheme for GPS trajectories. We confirm the effectiveness and robustness of our approach by considering a comprehensive set of attacks, i.e., adversarial trajectory modifications, including noise addition, point replacement, and size modifications. We conduct an

extensive evaluation using two real-world GPS trajectory datasets. We demonstrate that under the majority of considered attacks, *W-Trace* retains the watermark in 100% cases. We make our algorithm and data processing pipeline available as open source[1].

2 DEFINITIONS & PROBLEM FORMULATION

In this section, we introduce the definitions and the problem formulation, which we tackle with the proposed *W-Trace* approach.

Definition 2.1 (Trajectory). A trajectory T is a list of GPS coordinates ordered by the corresponding timestamps:

$$T = [(p_j, t_j)], \text{with } t_j < t_{j+1} \text{ for all } j,$$

where $p_j = (a_j, b_j)$ is the two-dimensional position with latitude a_j and longitude b_j and t_j is the timestamp of that position. Trajectory size, size(T), denotes the number of timestamps included in T.

A watermark is a signal embedded into the trajectory to enable verification of the trajectory origin. In this work, we represent watermarks as integer vectors.

Definition 2.2 (Watermark). A watermark $w \in \mathbb{Z}^m$ is an integer vector with the dimensionality m.

The dimensionality m of the watermark corresponds to the size of the (sub-)trajectory in which the watermark is embedded.

Watermark verification confirms if a given original watermark is embedded into the data and requires both the extracted watermark and the original watermark to be verified.

When the watermarking process modifies a trajectory T into $\widetilde{T}$, $\widetilde{T}$ needs to maintain usability for real-world applications. We make that intuition precise by defining a modification threshold.

Definition 2.3 (Modification threshold). A modification threshold σ bounds a distance D for trajectories. Given a modification threshold σ, we consider $\widetilde{T}$ a σ-modification of T if the spatial distance between these two trajectories is at most σ. Formally:

$$\mathrm{D}(T, \widetilde{T}) \leq \sigma. \tag{1}$$

In our experiments, we work with σ =10 meters, which reflects the typical inaccuracy of GPS sensors [1].

Our goal is to watermark GPS trajectories such that the watermarked trajectory remains usable for downstream applications and the watermark can be verified effectively, even if the watermarked trajectory is modified. Formally, given a watermark embedding procedure EMB, the respective watermarking verification procedure VER, and a watermark w, we aim that a trajectory T and its corresponding watermarked trajectory $\widetilde{T} = \mathrm{EMB}(T, w)$ obtained after applying watermarking are within the predefined modification threshold σ. Moreover, we aim that the verification of w with VER is possible, even if $\widetilde{T}'$ is modified from $\widetilde{T}$ within a modification threshold σ. Hence, we want to ensure that the verification $\mathrm{VER}(\widetilde{T}', T, W)$ returns true, if $\widetilde{T}'$ is a σ-modification of $\widetilde{T}$.

3 THE *W-TRACE* APPROACH

This section presents our proposed watermark embedding and verification method *W-Trace*.

[1]Software: https://github.com/Rajjat/watermarkingTrajectory

3.1 Watermark Embedding

Watermark embedding aims to incorporate a watermark into a given GPS trajectory. We consider a trajectory T of size n. We associate each GPS point (a_j, b_j) with a complex number,

$$c_j = a_j + ib_j, \tag{2}$$

where i is the imaginary unit. We split the transformed trajectory into multiple sub-trajectories of equal size. Next, we apply a Discrete Fourier Transform (DFT) to each sub-trajectory, where we use the Fast Fourier Transform (FFT) [8] algorithm for efficiency. DFT retrieves a frequency domain representation of the input and results in a sequence of complex numbers of the same length as the input. We feed the list of positions $c = (c_j)_{k \leq j < \ell}$ from the sub-trajectory spanning the indices k to ℓ, represented as complex numbers, into the FFT algorithm. The resulting frequency representations we then represent via amplitudes α and phase angles φ:

$$\alpha, \varphi \leftarrow \mathrm{FFT}(c). \tag{3}$$

Then, for a sub-trajectory, the watermark w with strength s is inserted in the amplitude α:

$$\widetilde{\alpha} = \alpha + s \cdot w. \tag{4}$$

A design decision of our method is to represent the watermark w as a vector of 1, −1, and 0 values of the same size as each sub-trajectory. This watermark is chosen and stored by the user; the watermark may be the same for each sub-trajectory or vary. In our experiments, we generate the watermarks randomly. The higher the watermark strength s, the more we modify the trajectory by inserting the watermark. In our experiments, we use $s = 0.0003$. We split each trajectory into sub-trajectories of size 16. In each sub-trajectory, we embed a watermark with 10 non-zero dimensions. Once the watermarks are inserted in the amplitude of each sub-trajectory, the next step is to apply an inverse FFT (IFFT) to obtain the watermarked sub-trajectory. We take the watermarked amplitude $\widetilde{\alpha}$ with the original phase φ and form a complex number $t_j \leftarrow \widetilde{\alpha}_j \exp(i\varphi_j)$. Applying the inverse FFT to the vector t, we obtain the watermarked trajectory $\widetilde{c}$. We abbreviate this as follows:

$$\widetilde{c} = (\widetilde{a}, \widetilde{b}) \leftarrow \mathrm{IFFT}(\widetilde{\alpha}, \varphi). \tag{5}$$

3.2 Watermark Extraction & Verification

Watermark verification aims to verify if the specific watermark w is embedded in the given trajectory $\widetilde{T}'$. This process includes four steps: selection of a candidate trajectory, trajectory size alignment, watermark extraction, and watermark correlation.

Candidate selection. As input, the watermark verification process requires the trajectory $\widetilde{T}'$ to be verified, the original trajectory T, the watermark w and the watermark strength parameter s adopted in the watermark embedding process. As the candidate original trajectory T, we select the closest user trajectory based on the minimum haversine distance to $\widetilde{T}'$.

Trajectory size alignment. Our watermark verification process requires T and $\widetilde{T}'$ to be of the same size. If $size(\widetilde{T}') > size(T)$, i.e. the trajectory size increased, we filter the coordinates from $\widetilde{T}'$ based on the minimum haversine distance to the candidate trajectory T. If the trajectory size of $\widetilde{T}'$ is smaller than $size(T)$, we fill the

positions in $\widetilde{T}'$ with a re-sampling of the closest point (regarding the haversine distance) to obtain the same size.

Watermark extraction. The watermark extraction process in *W-Trace* is non-blind, i.e., requiring the original data, and is the reverse of the watermark insertion process. We split $\widetilde{T}'$ into sub-trajectories of equal size and apply DFT to calculate the amplitude α'. We retrieve the watermark with:

$$w' = \frac{\alpha' - \alpha}{s}, \tag{6}$$

where α is the amplitude of the candidate trajectory T and s is the watermark strength.

Watermark correlation. The next step to verify the watermark is to compute the correlation between the extracted watermark w' and the original watermark w of each sub-trajectory. We adopt Normalized Cross-Correlation (NCC) – a widely used watermark verification measure [4]. NCC can successfully verify the watermarks in GPS trajectories, as demonstrated by our experiments. NCC of two watermarks, w and w', is computed as:

$$\mathrm{NCC}(w, w') = \frac{\sum_i w_i w'_i}{\sqrt{\sum_i w_i^2}\sqrt{\sum_i w_i'^2}}. \tag{7}$$

The value of NCC lies between -1 and 1. NCC value 1 indicates that two vectors are highly correlated, whereas 0 and -1 indicate no correlation and negative correlation, respectively. Finally, an average NCC score for all sub-trajectories of a given trajectory is calculated, and the verification is successful if this value is higher than the acceptance threshold τ. We adopt $\tau > 0.85$ based on [9].

4 THREAT MODEL: ATTACKS ON TRAJECTORIES

Digital watermarking is subject to adversarial attacks. The available knowledge limits the adversary's ability to prevent watermark verification. This paper assumes that an adversary has limited access, namely, knows the watermarked trajectory and the watermarking algorithm. In contrast, the original GPS data and the specific watermark embedded into the data remain unknown. An adversary with limited knowledge cannot remove the watermark directly. Instead, the adversary can attempt heuristic trajectory modifications to prevent watermark verification. We refer to such modifications as attacks on trajectories.

To quantify the utility of the trajectory modified in the adversarial settings for real-world applications, we follow the same principle as we introduced for the trajectory watermarking and apply a modification threshold σ:

$$\widetilde{T}' = AT(\widetilde{T}, \theta), \quad s.t.\ D(\widetilde{T}, \widetilde{T}') \leq \sigma.$$

Here, $AT(\cdot)$ is the attack function, $\widetilde{T}$ is the watermarked trajectory, θ represents the specific attack parameter, $D(\cdot)$ is the distance metric, $\widetilde{T}'$ is the modified watermarked trajectory, and σ is the modification threshold limiting the effects of the possible attacks on trajectories.

In this paper, we focus on the attacks discussed in the literature in the contexts of trajectory watermarking [9], trajectory similarity measures [10] and the more general perspective of cryptography [5]. In particular, we consider four different attack types: noise additive attacks, point replacement attacks, size modification attacks, and the combination of these types, the hybrid attack.

Noise Additive Attacks. In noise additive attacks, noise is inserted into trajectory coordinates.

(1) **Additive Gaussian White Noise (AGWN)** In this attack, for each position in the trajectory, a random sample from a normal distribution is drawn and added to the GPS position.
(2) **Additive Signal to Noise Ratio (ASNR)** This attack is similar to the previous attack, but we scale the noise to achieve a selected signal-to-noise ratio (SNR).
(3) **Additive Outliers with SNR (AOSNR)** We randomly select points with the probability $\theta = (p_{\mathrm{AOSNR}})$, and then add scaled noise to these positions.
(4) **Double Embedding Attack (DEA)** In the double embedding attack, an adversary attempts to remove the original watermark by embedding a different watermark with the same approach as the original watermark.

Point Replacement Attacks. Point replacement attacks remove specific trajectory elements and replace them with information based on the adjacent points.

(1) **Replace Random Points (RRP)** Points are selected with the probability $\theta = (p_{\mathrm{RRP}})$, and then those selected points are replaced with their respective previous points.
(2) **Replace Random Points with Path (RRPP)** replaces each point with the probability $\theta = (p_{\mathrm{RRPP}})$. The replaced value is a convex combination of the remaining adjacent points.
(3) **Replace Non-Skeleton Points with Path (RNSPP)** In this attack, we use the Ramer–Douglas–Peucker (RDP) algorithm. The points removed by the RDP algorithm are replaced with a convex combination of the adjacent points.

Size Modification Attacks. In size modification attacks, the trajectory size is modified either by cropping or interpolation.

(1) **Linear Interpolation Attack (LIA)** Additional points are inserted at random positions in the trajectory by linear interpolation, increasing the trajectory size.
(2) **Cropping Attack (CA)** Cropping attack removes selected points from the trajectory, decreasing the trajectory size.

Hybrid Attacks. An adversary can combine several attacks on the same trajectory. We exemplify a hybrid attack as a sequence of a cropping attack (CA) followed by additive Gaussian white noise (AGWN) and replace random points (RRP).

5 EVALUATION

We aim to evaluate the effectiveness and robustness of *W-Trace* regarding the threat model. In this section, we describe the experimental setup and results.

Datasets. We use two real-world trajectory datasets for evaluating the proposed watermarking method. We randomly selected 1100 trajectories of size 256 from each dataset.

(1) **German Dataset** is provided by a proprietary data provider. The dataset contains trajectory data of vehicles from two German federal states: Saxony and Lower Saxony, in September 2019. The average sampling rate is 12 times per minute.

Table 1: Recognition Rate of *W-Trace* and baseline methods on the German and Porto datasets.

		Noise additive				Point replacement			Size mod.		Hybrid	Avg.
Method	Dataset	AGWN	ASNR	AOSNR	DEA	RRP	RNSPP	RRPP	LIA	CA		
SVD	German	100.0	79.4	99.7	0.0	100.0	65.3	100.0	100.0	100.0	100.0	84.3
(Blind)	Porto	100.0	98.2	99.3	0.0	100.0	94.7	100.0	100.0	100.0	100.0	89.2
IMF	German	72.5	70.6	74.5	75.2	75.8	76.0	75.1	76.0	77.1	72.1	74.5
(Non-blind)	Porto	87.2	87.0	90.3	90.8	90.1	90.8	90.7	90.3	91.0	87.1	89.5
TrajGuard	German	87.6	83.2	94.4	94.4	95.6	74.2	95.9	75.2	91.9	83.8	87.6
(Blind)	Porto	59.8	56.2	55.7	61.7	68.3	65.0	68.3	63.6	64.5	57.5	62.1
W-Trace	German	100.0	99.8	98.2	100.0	98.6	100.0	100.0	100.0	100.0	94.0	99.0
(Non-blind)	Porto	100.0	100.0	99.0	100.0	100.0	100.0	100.0	100.0	100.0	99.9	99.8

(2) **Porto Dataset** contains variable size trajectories generated by 442 taxis from July 1, 2013, to June 30, 2014, in Porto, Portugal [7]. The sampling rate is four times per minute.

Baselines. We adopt state-of-the-art watermarking methods from the audio domain and GPS trajectories domain.

(1) **IMF Watermarking [4]** is a non-blind technique used in watermarking audio signals. Each trajectory is represented as a signal (latitude/longitude vs. time) and decomposed into multiple parts using Empirical Mode Decomposition (EMD).
(2) **TrajGuard [9]** watermarks a GPS trajectory using a geometric transformation based on a blind scheme, i.e., it does not require the original data for the extraction. TrajGuard partitions the trajectory into multiple parts and then distributes the watermark into all the sub-trajectories.
(3) **SVD Watermarking [2]** is based on a blind audio watermarking scheme. This method uses Singular Value Decomposition (SVD) and quantization index modulation.

Evaluation Metrics. To assess the watermark verification effectiveness and robustness, i.e., the ability to correctly recognize a watermark in modified trajectory data, we adopt **recognition rate**. Recognition rate is the ratio of the number of correctly identified watermarked trajectories (true positives, TP) to the total number of watermarked trajectories: $\text{Recognition rate} = TP/(TP + FN)$, where FN is the number of false negatives, i.e., unrecognized watermarked trajectories. Following [9], we accept the watermark to be successfully verified if the average watermark correlation between the noised trajectory and watermarked trajectory is higher than the acceptance threshold, i.e., $\tau > 85\%$.

Evaluation Results. *W-Trace* approach is effective and robust against all the considered attacks in both datasets, as shown in Table 1. The average recognition rate of *W-Trace* is around 99% in both datasets, confirming the effectiveness, robustness, and generalizability of *W-Trace*. Baseline methods demonstrate varying performance against some attacks across the two datasets. For example, TrajGuard does not perform well in multiple attacks, especially on the Porto dataset. This is because the Porto dataset is spatially denser than the German dataset, making TrajGuard more vulnerable to attacks [9]. Furthermore, TrajGuard embeds a smaller amount of watermark information, leading to a lower recognition rate. IMF watermarking failed to detect the watermark in the German dataset, whereas this method works well for the Porto dataset. The German dataset covers a large geographical area, including two German federal states, whereas the Porto dataset is limited to one city. A denser spatial area of the Porto dataset leads to a better decomposition and makes the verification process more effective. Regarding the SVD watermarking, we observe that the DEA attack destroys the quantization-based watermark detection process. In summary, in contrast to the baselines, *W-Trace* is more robust against the considered attacks and less dependent on data sparsity.

ACKNOWLEDGMENTS

This work is partially funded by the Federal Ministry for Economic Affairs and Climate Action (BMWK), Germany, under "CampaNeo" (01MD19007B), and "d-E-mand" (01ME19009B), the European Commission (EU H2020) under "smashHit" (871477), the German Research Foundation under "WorldKG" (424985896), and by the B-IT foundation and the state of North Rhine-Westphalia (Germany).

REFERENCES

[1] David M Bevly. 2004. Global positioning system (GPS): A low-cost velocity sensor for correcting inertial sensor errors on ground vehicles. *Journal of dynamic systems, measurement, and control* (2004), 255–264.
[2] Vivekananda Bhat, Indranil Sengupta, and Abhijit Das. 2011. A new audio watermarking scheme based on singular value decomposition and quantization. *Circuits, Systems, and Signal Processing* (2011), 915–927.
[3] Chi-Yin Chow and Mohamed F Mokbel. 2011. Trajectory privacy in location-based services and data publication. *ACM SIGKDD Explorations Newsletter* (2011), 19–29.
[4] Basant S. Abd El-Wahab, Heba Ali El-Khobby, Mustafa M. Abd-Elnaby, and Fathi E. Abd El-Samie. 2021. Simultaneous speaker identification and watermarking. *International Journal of Speech Technology* (2021), 205–218.
[5] Raju Halder, Shantanu Pal, and Agostino Cortesi. 2010. Watermarking Techniques for Relational Databases: Survey, Classification and Comparison. *J. UCS* (2010), 3164–3190.
[6] Xiaoming Jin, Zhihao Zhang, Jianmin Wang, and Deyi Li. 2005. Watermarking Spatial Trajectory Database. In *DASFAA*. 56–67.
[7] Luis Moreira-Matias, Michel Ferreira, Joao Mendes-Moreira, L. L., and J. J. 2015. Taxi Service Trajectory - Prediction Challenge, ECML PKDD 2015. UCI Machine Learning Repository.
[8] Henri J Nussbaumer. 1981. The fast Fourier transform. In *Fast Fourier Transform and Convolution Algorithms*. 80–111.
[9] Zheyi Pan, Jie Bao, Weinan Zhang, Yong Yu, and Yu Zheng. 2019. TrajGuard: A Comprehensive Trajectory Copyright Protection Scheme. In *25th ACM SIGKDD*. 3060–3070.
[10] Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. 2020. A survey of trajectory distance measures and performance evaluation. *The VLDB Journal* (2020), 3–32.