Privacy-preserving Exemplar-based Top-k Spatial Dataset Search in Cloud

Xin Li
School of Computer Science, Nanjing University of Posts and Telecommunications
Nanjing, China
1024041115@njupt.edu.cn

Abstract

The development of cloud computing has met the growing demand for dataset search in the era of massive data. In the field of spatial dataset search, the high prevalence of sensitive information highlights the need for privacy-preserving search processing in the cloud. However, existing ciphertext search schemes for spatial datasets are not effective enough. In this paper, We present a privacy-preserving spatial dataset search scheme that addresses these challenges. A density distribution-based model is first employed to quantify spatial similarity between datasets, which is then coupled with the AME encryption scheme to enable secure relative similarity comparison. With these core components integrated, we propose the baseline Priekds search scheme. For improved search efficiency, we develop an AME-based chunk encryption method that powers the optimized Priekds+ scheme. The security of both schemes receives rigorous verification through game simulation proofs. Experimental evaluation across three spatial data repositories confirms the effectiveness and efficiency of our approach.

CCS Concepts

• **Security and privacy → Management and querying of encrypted data**; • **Information systems → Information retrieval query processing**.

Keywords

Cloud Computing, Spatial Dataset, Dataset Search, Data Privacy

ACM Reference Format:
Xin Li. 2025. Privacy-preserving Exemplar-based Top-k Spatial Dataset Search in Cloud. In . ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

With the rapid development of artificial intelligence and big data technologies, datasets enhance their value in science, business and technology, driving the need for efficient dataset search capabilities[4, 5]. In this context, cloud computing becomes a key factor in providing powerful computing and storage capabilities for large-scale datasets[2, 9]. Many organizations choose to deploy dataset search services in the cloud to efficiently manage and search datasets. Among these datasets, spatial datasets hold unique significance[14]. With sensitive geolocation information, they support a variety of critical applications ranging from urban route planning [25] to disaster management [19]. Outsourcing data to the cloud, although it enhances data accessibility, presents a significant challenge by removing data owners' direct control over their information. This can lead to security issues such as potential breaches and unauthorized access[6, 12, 23]. Addressing these concerns, directly encrypting spatial datasets is a strategy to reduce risks. However, this method reduces data usability and can decrease the efficiency of dataset search services. Consequently, it is crucial to create a search scheme for spatial datasets that can support both privacy preservation and dataset search services.

In the spatial dataset search problem, we focus on exemplar-based spatial dataset search [16, 17, 24]. Exemplar datasets allow users to search for spatial datasets by providing a sample dataset of interest as a search request and returning the k spatial datasets that are most similar to the example dataset as search results. For example, a public health official may search for spatial datasets related to the spread of infectious diseases in certain areas[1]. By analyzing historical outbreaks with similar geographic distribution, similar transmission patterns and trends can be detected, allowing for the prediction of possible transmission paths for current outbreaks and the development of prevention and control measures in advance. Existing research has proposed a number of schemes for measuring the similarity of spatial datasets. One way is to calculate the size of the overlap area between the minimum bounding rectangles (MBRs) that enclose all points in each dataset[7, 8, 10]. Another way is to calculate the earth mover's distance (EMD)[3], which is used to measure the similarity between datasets by transforming datasets into distributions. In addition, Hausdorff distance is used to calculate the maximum distance between the nearest neighbour points in spatial datasets[18, 21, 22], etc. Existing spatial dataset similarity metrics have limitations.MBR only considers the overall extent of the data and ignores the internal data distribution; Hausdorff Distance is sensitive to outliers; and EMD has high computational complexity. Moreover, none of these methods fully consider privacy protection, especially in cloud environments, and how to design privacy-preserving spatial dataset search schemes remains an urgent challenge.

In this paper, we propose privacy-preserving spatial dataset search schemes. In the proposed schemes, a density distribution-based similarity model (DDSM) is designed to measure the similarity between spatial datasets, which converts spatial datasets into density distributions for measuring similarity. Then, we vectorize the density distribution with some variations and encryption for

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

secure similarity computation. Based on the above ideas, we propose a baseline search scheme (Priekds). In addition, we design an optimized search scheme (Priekds+) with chunked encryption to improve the search efficiency. Furthermore, a game simulation-based proof analysis verifies the security of the proposed schemes, while experimental results demonstrate their accuracy and efficiency.

Overall, the contributions of this paper are as follows:

- We propose a novel density distribution-based similarity model, which converts spatial datasets into density distributions for similarity calculation. (see Section 4)
- We propose a baseline privacy-preserving exemplar-based Top-k spatial dataset search scheme Priekds, which enables the privacy-preserving exemplar-based Top-k spatial dataset search. (see Section 5)
- We design a chunked encryption algorithm and based on it we propose an optimized search scheme Priekds+ which significantly improves the search efficiency. (see Section **??**)
- We present a game simulation-based proof to analyze the security of the proposed schemes and conduct comprehensive experiments on three real spatial data repositories. (see Section **??** and Section 6)

2 Related Work

Privacy-preserving spatial data search. Various schemes have been proposed to enable privacy-preserving spatial data search in the cloud, e.g., Xu et al. [27] employed secure KNN computation, polynomial fitting technique, and order-preserving encryption to achieve secure geometric range query over cloud data. Miao et al. [15] combined Geohash and asymmetric scalar-product-preserving encryption (ASPE) [26] to use the vector inner product to determine whether spatial data meets search requirements. Li et al. [13] design the order-preserving encrypted similarity to achieve secure similarity calculation . Zheng et al. [30] proposes a novel Asymmetric Matrix Encryption (AME) scheme that can securely perform Euclidean distance computation and one-sided non-interactive distance comparison. .

Spatial dataset search. The key in spatial dataset search is to measure the similarity between datasets. Commonly used measures include the Minimum Bounding Rectangle (MBR) [10], the Hausdorff distance [20, 28] and the Earth Mover's Distance (EMD) [29], et al. For example, Vasconcelos et al. [7] measured the spatial dataset similarity based on the overlapping area of the bounding boxes encompassing all data. Nutanong et al. [18] used the Hausdorff distance as a criterion to measure the similarity between two spatial datasets. Yang et al. [29] tackled spatial dataset search by using EMD to measure the similarity of spatial datasets.

3 Models and Problem Formulation

3.1 System Model and Threat Model

System Model. In this paper, we consider such a system model, as shown in Figure 1, which is composed of three key elements: Data Owner (DO), Data User (DU), and Cloud Server (CS). The details are as follows:

- DO is in charge of encrypting the spatial datasets and uploading them to CS.
- DU converts the search instructions into trapdoors and sends them to CS to perform the search processing.
- CS has strong computing and storage capabilities, performs the search process according to the trapdoor, and returns the search results to DU.

Threat Model. The threat model is "curious but honest" [**?**], win which the CS honestly performs data storage and search operations but may try to deduce sensitive information from encrypted data and queries. DO encrypt their spatial datasets before outsourcing them to the CS, while DU generate trapdoors to submit search requests, guaranteeing that all computations are executed on ciphertexts.

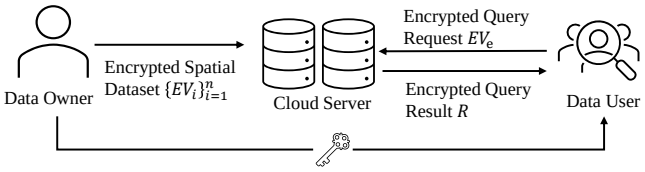

Figure 1: System model and threat model

3.2 Security Definitions

We adopt a game simulation [11] to define our schemes' security. Simplifying, we denote Π as a privacy-preserving spatial dataset search scheme, and $\mathcal{A}$ and $\mathcal{S}$ as the adversary and the simulator, respectively. Specifically, we design two games involving $\mathcal{A}$, namely $Real_{\mathcal{A}}^{\Pi}(\lambda)$ and $Ideal_{\mathcal{A}}^{\Pi}(\lambda)$. In $Real_{\mathcal{A}}^{\Pi}(\lambda)$, a real system performs spatial dataset search processing on encrypted datasets, whereas in $Ideal_{\mathcal{A}}^{\Pi}(\lambda)$, a simulated system allows $\mathcal{S}$ to obtain the search result using leaked information. If the output results of $Real_{\mathcal{A}}^{\Pi}(\lambda)$ and $Ideal_{\mathcal{A}}^{\Pi}(\lambda)$ are indistinguishable when the search requests are the same, the scheme is considered to be $\mathcal{F}$-adaptively secure, where $\mathcal{F}$ is a set of leakage functions. The set of leakage functions for the scheme Π is denoted as $\mathcal{F}_{\Pi}$, and its definition will be provided in Section **??**. The security definition is presented as follows:

Definition 1. $\mathcal{F}$-adaptively Secure. Given a scheme Π, if for an adversary $\mathcal{A}$, there is a simulator $\mathcal{S}$ with a set of leakage functions $\mathcal{F}_{\Pi}$ for Π such that Eq.(**??**) holds, then Π is $\mathcal{F}$-adaptively secure.

3.3 Problem Formulation and Design Goals

A spatial data repository, denoted as $\mathcal{D} = \{D_1, D_2, \cdots, D_n\}$, is a collection of spatial datasets. Each D_i within the repository comprises a collection of spatial points, where each spatial point is represented by a two-dimensional coordinate.

Definition 2. Exemplar-based Top-k Spatial Dataset Search. Given an exemplar dataset D_e, the exemplar-based top-k spatial dataset search, denoted as $Q = (\mathcal{D}, D_e, k)$, is to obtain the k most similar spatial datasets from $\mathcal{D}$ to D_e. The search result R should satisfy the condition,

$$|R| = k \wedge \forall D_i \in R, D_j \notin R(Sim(D_i, D_e) > Sim(D_j, D_e)),$$

where $Sim(D_i, D_e)$ is the similarity between D_i and D_e.

This paper intends to develop privacy-preserving exemplar dataset search schemes capable of effectively conducting the search for exemplar datasets while safeguarding the privacy of the datasets and the search requests.

4 Density Distribution-based Spatial Dataset Similarity Model

In this section, we propose a density distribution-based similarity model (DDSM), which first converts datasets into density distributions and then performs similarity calculations on them.

4.1 Density Distribution of Spatial Dataset

In this section, we propose a density distribution-based similarity model (DDSM), which initially transforms spatial datasets into density distributions, and subsequently calculates the similarities among these distributions.

Definition 3. Grid-based Space Representation. Given a domain of 2-dimensional spatial space $\mathcal{S}$ and a grid partition threshold θ, $\mathcal{S}$ is equally divided into $\theta \times \theta$ grids. Each grid is uniquely identified by its integer coordinates $g_{(x,y)}$, where the grid is in x-th row and y-th column, and thus $\mathcal{S}$ can be represented by a set of grid coordinates, i.e., $\mathcal{S} = \{g_{(x,y)} \mid x, y \in \mathbb{Z}, 0 \le x, y < \theta\}$

Definition 4. Spatial Dataset Density Distribution. Given a spatial dataset D_i , the density distribution of D_i is a set of pairs, which is denoted as

$$G_i = \{g_{i,(x,y)}, \rho_{i,(x,y)} \mid x, y \in \mathbb{Z}, 0 \le x, y < \theta\}, \quad (1)$$

where $g_{i,(x,y)}$ is the ID of a grid and $\rho_{i,(x,y)}$ is the density of $g_{i,(x,y)}$. The calculation of $\rho_{i,(x,y)}$ is $\rho_{i,(x,y)} = \frac{N_{i,(x,y)}}{|D_i|}$, where $N_{i,(x,y)}$ is the number of points of D_i in $g_{i,(x,y)}$ and $|D_i|$ is the total number of points in D_i. which guarantees that $\rho_{i,(x,y)}$ has a value between 0 and 1.

We use Figure 2 to illustrate the above definitions. The space is divided into 4×4 grids and each grid has a unique ID, i.e., $\mathcal{S} = \{(0,0), (0,1), \cdots, (3,3)\}$. There are two spatial datasets D_i and D_j located in $\mathcal{S}$. The spatial dataset density distributions G_i and G_j can be obtained by calculating the density of each non-empty grid.

4.2 Density Distribution-based Similarity

The relatively small area of each grid allows points located within the same grid to be close to each other. If points from different datasets fall in the same grid, these points have some degree of similarity within that grid. Based on this local similarity, we can measure the overall similarity between two datasets by calculating the sum of the squares of the density differences between their density distributions in each grid.

Definition 5. Density Distribution-based Spatial Dataset Similarity. Given the density distribution G_i of D_i and G_j of D_j, the similarity between D_i and D_j is denoted as $Sim(G_i, G_j)$, which is calculated follows

$$Sim(G_i, G_j) = 1 - \sqrt{\frac{\sum_{(x,y)\in S}(\rho_{i,(x,y)} - \rho_{j,(x,y)})^2}{2}} \quad (2)$$

According to Eq.(2), the similarity measure for the spatial dataset considers the full grid in the space $\mathcal{S}$. For each grid $g_{i,(x_\alpha,y_\alpha)} = g_{j,(x_\beta,y_\beta)}$, calculate the square of the density difference between the two datasets in this grid $(\rho_{i,(x_\alpha,y_\alpha)} - \rho_{j,(x_\beta,y_\beta)})^2$, and accumulate these values,then apply normalization to the accumulated value and subtract that normalized value from 1 to measure the similarity between the two spatial datasets.

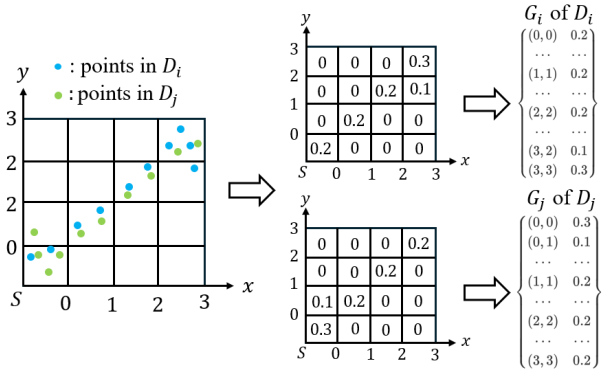

Figure 2: An example of spatial dataset density distribution

We also use Figure 2 to illustrate Definition 5. According to Eq.(3), $\sum_{(x,y)\in S}(\rho_{i,(x,y)} - \rho_{j,(x,y)})^2 = (0.2-0.3)^2 + (0-0.1)^2 + (0.2-0.2)^2 + (0.2-0.2)^2 + (0.1-0)^2 + (0.3-0.2)^2 = 0.01 + 0.01 + 0 + 0 + 0.01 + 0.01 = 0.04$. Thus, we can calculate the similarity, $Sim(G_i, G_j)$, between spatial datasets D_i and D_j. $Sim(G_i, G_j) = 1 - \sqrt{\frac{0.04}{2}} = 0.86$.This means that D_i and D_j have an 86% similarity in terms of their spatial distribution, indicating a high degree of consistency between the two datasets.

In summary, a similarity model based on density distribution can characterize the distribution of a spatial dataset by dividing the space into a grid. Compared with MBR, it describes the dataset in more detail and does not ignore the internal details of the minimum bounding rectangle. It reduces computational complexity compared to EMD. It accurately responds to the differences in spatial datasets in each grid compared to the similarity comparison method of histogram intersection. In practice, if the space is divided into smaller grids, the distance between points in the same grid will be closer and the measurements between data sets will be finer. However, this will increase the computational cost of similarity calculation for spatial datasets and may lead to a significant increase in the proportion of zero values in the stored data, which will result in an increase in the sparsity of the data. Therefore, it is necessary to choose the appropriate resolution according to the actual situation. We will conduct experiments on the effect of resolution in this section 6.

4.3 Vectorization and Extension of Density Values for Spatial Datasets

In Priekds, DDSM is used to calculate the similarity between spatial datasets. Because the spatial dataset density distribution reflects the location information and statistical information of datasets, they are sensitive information and need to be processed for privacy preservation in search processing.

We use a vectorization approach to convert the density values of each grid in the spatial dataset into an ordered vector form, i.e., a spatial dataset density vector (SD-vector), based on the ID of the grid.

Definition 6. Spatial Dataset Density Vector (SD-vector). Given a density distribution G_i of spatial dataset D_i partitioned into $\theta \times \theta$ grids, let $V_i^{(1)}$ denote the θ^2-dimensional density vector obtained by column-major flattening. The SD-vector V_i extends $V_i^{(1)}$ to dimension $\theta^2 + 1$:

$$V_i[h] = \begin{cases} V_i^{(1)}[h] & \text{if } 0 \le h < \theta^2, \\ -0.5\|V_i^{(1)}\|^2 & \text{if } h = \theta^2, \end{cases} \tag{3}$$

where $\|V_i^{(1)}\|^2 = \sum_{h=0}^{\theta^2-1}(V_i^{(1)}[h])^2$ is the squared Euclidean norm.

Definition 7. Spatial Dataset Density Pair (SD-pair). Given an SD-vector V_i, its SD-pair $(V_{i,L}, V_{i,R})$ consists of two $2(\theta^2+1)$-dimensional vectors:

$$V_{i,L} = \begin{bmatrix} V_i \\ -\mathbf{1} \end{bmatrix}, \quad V_{i,R} = \begin{bmatrix} \mathbf{1} \\ V_i \end{bmatrix},$$

where $\mathbf{1}$ is the (θ^2+1)-dimensional all-ones vector.

Definition 8. Exemplar Dataset Density Vector (ED-vector). Give an exemplar dataset density distribution G_e of exemplar dataset D_e, its ED-vector V_e is constructed analogously to V_i but with:

$$V_e[h] = \begin{cases} V_e^{(1)}[h] & \text{if } 0 \le h < \theta^2, \\ 1 & \text{if } h = \theta^2. \end{cases} \tag{4}$$

Definition 9. Spatial Exemplar Dataset Density Matrix (ED–matrix). Given the ED-vector of spatial exemplar dataset V_e, let $\mathrm{Diag}(V_e)$ denote the $(\theta^2+1) \times (\theta^2+1)$ diagonal matrix whose main diagonal entries are the elements of V_e. The ED-matrix M_e is a block-diagonal matrix of dimension $2(\theta^2+1) \times 2(\theta^2+1)$ defined by:

$$M_e = \begin{bmatrix} \mathrm{Diag}(V_e) & 0 \\ 0 & \mathrm{Diag}(V_e) \end{bmatrix},$$

where O is the $(\theta^2+1) \times (\theta^2+1)$ zero matrix.

Lemma 1. Given two spatial datasets D_i and D_j and an exemplar spatial dataset D_e,and thier density distribution G_i,G_j and G_e, we have

$$V_{i,L}^T \cdot M_e \cdot V_{j,R} > 0 \Leftrightarrow Sim(G_i, G_e) > Sim(G_j, G_e)$$

Proof: According to Definition 7 and 9, we can deduce:

$$V_{i,L}^T \cdot M_e \cdot V_{j,R} = \begin{bmatrix} V_i^T \\ -\mathbf{1} \end{bmatrix}^T \begin{bmatrix} \mathrm{diag}(V_e) & O \\ O & \mathrm{diag}(V_e) \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ V_j^T \end{bmatrix},$$

$$= V_i^T \mathrm{Diag}(V_e)\mathbf{1} - \mathbf{1}^T \mathrm{Diag}(V_e) V_j$$

$$= \sum_{h=0}^{\theta^2-1} V_i^{(1)}[h] V_e^{(1)}[h] - \sum_{h=0}^{\theta^2-1} V_e^{(1)}[h] V_j^{(1)}[h] - 0.5\|V_i^{(1)}\|^2 + 0.5\|V_j^{(1)}\|^2$$

$$= \sum_{(x,y)} (\rho_{j,(x,y)} - \rho_{e,(x,y)})^2 - \sum_{(x,y)} (\rho_{i,(x,y)} - \rho_{e,(x,y)})^2$$

$$= \mathrm{Sim}(G_i, G_e) - \mathrm{Sim}(G_j, G_e)$$

and it follows that $V_{i,L}^T \cdot M_e \cdot V_{j,R} > 0$,

$\Rightarrow \mathrm{Sim}(G_i, G_e) > \mathrm{Sim}(G_j, G_e)$.

According to the above derivations, Lemma 1 holds. ∎

Lemma 1 indicates that whether the product of $V_{i,L}^T \cdot M_e \cdot V_{j,R}$ is greater than 0 can indicate the size relationship of the similarity between these two spatial datasets and the exemplar dataset.

5 Baseline Search Scheme

In this section, we present the privacy-preserving exemplar-based top-k spatial dataset baseline search scheme (Priekds). We present vectorization of the density distribution of spatial datasets as well as strategies for computing similarity and comparisons in an encrypted environment.

Definition 10. Transformed Spatial Sample Density Matrix (TSSD-matrix). Given the exemplar dataset D_e's ED-matrix M_e, the generation of new TED-matrix involves incorporating several random numbers defined as follows $\{\widehat{z}_e, \widetilde{z}_e, \widehat{\alpha}_e, \widehat{\beta}_e, \widetilde{\alpha}_e, \widetilde{\beta}_e\} \in \mathbb{R}$ and these random numbers must satisfy the following conditions:

$$\begin{cases} \widehat{\alpha}_e > \widehat{\beta}_e > 0 \\ \widetilde{\alpha}_e > \widetilde{\beta}_e > 0 \end{cases}$$

Additionally, the random numbers $\widehat{\gamma}_e$ and $\widetilde{\gamma}_e$, must belong to the set of real numbers $\mathbb{R}$ and satisfy the equation $\widehat{\gamma}_e + \widetilde{\gamma}_e = 0$. Under these conditions, two matrices, denoted as $\widehat{M_e}$ and $\widetilde{M_e}$.

$$\widehat{M_e} = \begin{bmatrix} \widehat{\alpha}_e M_e & 0 & 0 & 0 & 0 \\ 0 & \widehat{z}_e & 0 & 0 & 0 \\ 0 & 0 & \widehat{z}_e & 0 & 0 \\ 0 & 0 & 0 & \widehat{\beta}_e & 0 \\ 0 & 0 & 0 & 0 & \widehat{\gamma}_e \end{bmatrix},$$

$$\widetilde{M_e} = \begin{bmatrix} \widetilde{\alpha}_e M_e & 0 & 0 & 0 & 0 \\ 0 & \widetilde{z}_e & 0 & 0 & 0 \\ 0 & 0 & \widetilde{z}_e & 0 & 0 \\ 0 & 0 & 0 & \widetilde{\beta}_e & 0 \\ 0 & 0 & 0 & 0 & \widetilde{\gamma}_e \end{bmatrix}$$

The TED-matrix is consequently expressed as the ordered pair $TM_e = (\widehat{M_e}, \widetilde{M_e})$.

Definition 11. Transformed Spatial Density Vector (TSD-vector). Given the SD-pair $(V_{i,L}, V_{i,R})$ of dataset D_i, the generation of TD-pair involves incorporating several random numbers defined as follows $\widehat{z}_L, \widehat{z}_R, \widetilde{z}_L, \widetilde{z}_R, \widehat{\alpha}_L, \widehat{\alpha}_R, \widehat{\beta}_L, \widehat{\beta}_R, \widetilde{\alpha}_L, \widetilde{\alpha}_R, \widetilde{\beta}_L, \widetilde{\beta}_R \in \mathbb{R}$, and these random numbers must satisfy the following conditions:

$$\begin{cases} \widehat{\alpha}_L > \widehat{\beta}_L > 0, \quad \widehat{\alpha}_R > \widehat{\beta}_R > 0 \\ \widetilde{\alpha}_L > \widetilde{\beta}_L > 0, \quad \widetilde{\alpha}_R > \widetilde{\beta}_R > 0 \end{cases}$$

Using these parameters, four transformed vectors are constructed:

$$\widehat{V}_{i,L} = \begin{bmatrix} \widehat{\alpha}_L V_{i,L} \\ \widehat{z}_L \\ \widehat{z}_L \\ \widehat{\beta}_L \end{bmatrix}, \widehat{V}_{i,R} = \begin{bmatrix} \widehat{\alpha}_R V_{i,R} \\ \widehat{z}_R \\ -\widehat{z}_R \\ \widehat{\beta}_R \end{bmatrix}, \widetilde{V}_{i,L} = \begin{bmatrix} \widetilde{\alpha}_L V_{i,L} \\ \widetilde{z}_L \\ \widetilde{z}_L \\ \widetilde{\beta}_L \end{bmatrix}, \widetilde{V}_{i,R} = \begin{bmatrix} \widetilde{\alpha}_R V_{i,R} \\ \widetilde{z}_R \\ -\widetilde{z}_R \\ \widetilde{\beta}_R \end{bmatrix}.$$

The TD-pair TP_i for D_i is then defined as the quadruple:

$$TP_i = \left(\widehat{V}_{i,L}, \widehat{V}_{i,R}, \widetilde{V}_{i,L}, \widetilde{V}_{i,R}\right).$$

We will further encrypt the vectors to achieve secure spatial dataset similarity comparison.

5.1 Encryption on Vectorised Density Distribution

To enhance the privacy protection of TSD-vectors and TSSD-vectors, we encrypt these vectors using the encryption scheme AME. The encryption algorithm $Enc_n(\cdot)$ and $Enc_e(\cdot)$ are shown in the algorithm 1 and algorithm 1 respectively, which requires some random invertible matrices as the private key. We use the key to encrypt the vectors: The keys of TSD-vectors and TSSD-matrices are defined as:

$$sk = \left\{ \widehat{M}_L^{u,v,w}, \widehat{M}_R^{u,v,w}, \widetilde{M}_L^{u,v,w}, \widetilde{M}_R^{u,v,w} \mid u,v,w \in \{1,2\} \right\},$$

where sk consists of eight random $2\theta^2+6$ invertible matrices. Using the above keys, we present the encryption on the TSD-vectors, shown in Definition 12 .

Algorithm 1 $Enc_n(TV_i, sk)$

Input: Vector set TV_i, secret keys sk
Output: Encrypted vector EV_i

Step 1: Vector Decomposition

$$\widehat{V}_{i,L} = \widehat{V}_{i,L,1} + \widehat{V}_{i,L,2} \quad \widetilde{V}_{i,L} = \widetilde{V}_{i,L,1} + \widetilde{V}_{i,L,2}$$
$$\widehat{V}_{i,R} = \widehat{V}_{i,R,1} + \widehat{V}_{i,R,2} \quad \widetilde{V}_{i,R} = \widetilde{V}_{i,R,1} + \widetilde{V}_{i,R,2}$$

Step 2: Component Encryption for $u \leftarrow 1$ **to** 2 **do**
  for $v \leftarrow 1$ **to** 2 **do**
    for $w \leftarrow 1$ **to** 2 **do**
      $\widehat{EV}_{i,L}^{u,v,w} = \widehat{V}_{i,L,u}^T \widehat{M}_L^{u,v,w}$
      $\widetilde{EV}_{i,L}^{u,v,w} = \widetilde{V}_{i,L,u}^T \widetilde{M}_L^{u,v,w}$
      $\widehat{EV}_{i,R}^{u,v,w} = (\widehat{M}_R^{u,v,w})^{-1} \widehat{V}_{i,R,w}$
      $\widetilde{EV}_{i,R}^{u,v,w} = (\widetilde{M}_R^{u,v,w})^{-1} \widetilde{V}_{i,R,w}$
    end
  end
end

Step 3: Result Aggregation

$$EV_i = \left\{ \widehat{EV}_{i,L}^{u,v,w}, \widehat{EV}_{i,R}^{u,v,w}, \widetilde{EV}_{i,L}^{u,v,w}, \widetilde{EV}_{i,R}^{u,v,w} \;\middle|\; u,v,w \in \{1,2\} \right\}$$

return EV_i

Definition 12. Encrypted TSD-vector of Spatial Dataset. Given a TSD-vector $TV_i = \{\widehat{V}_{i,L}, \widehat{V}_{i,R}, \widetilde{V}_{i,L}, \widetilde{V}_{i,R}\}$ for dataset D_i, its encrypted TSD-vector EV_i is generated by:

$$EV_i = Enc_n(TV_i, sk) = \left\{ \begin{matrix} \widehat{V}_{i,L,u}^T \widehat{M}_L^{u,v,w}, \ (\widehat{M}_R^{u,v,w})^{-1} \widehat{V}_{i,R,w}, \\ \widetilde{V}_{i,L,u}^T \widetilde{M}_L^{u,v,w}, \ (\widetilde{M}_R^{u,v,w})^{-1} \widetilde{V}_{i,R,w} \end{matrix} \;\middle|\; u,v,w \in \{1,2\} \right\} \tag{5}$$

where Enc_n is the encryption algorithm defined in Algorithm 1. Given a dataset $\mathcal{D} = \{D_i\}_{i=1}^n$ with corresponding TSD-vectors $\{TV_i\}_{i=1}^n$, the encrypted dataset is obtained by:

$$\mathcal{EV} = \{Enc_n(TV_i, sk) \mid i = 1, \ldots, n\} = \{EV_i\}_{i=1}^n$$

Algorithm 2 $Enc_e(TV_e, sk)$

Input: Trapdoor vectors $TV_e = \{\widehat{V}_e', \widetilde{V}_e'\}$, secret keys sk
Output: Encrypted matrices EV_e

Step 1: Vector Decomposition

$$\widehat{V}_e' \rightarrow \{\widehat{V}_{e,1}, \widehat{V}_{e,2}\} \quad \text{s.t. } \widehat{V}_{e,1} + \widehat{V}_{e,2} = \widehat{V}_e'$$
$$\widetilde{V}_e' \rightarrow \{\widetilde{V}_{e,1}, \widetilde{V}_{e,2}\} \quad \text{s.t. } \widetilde{V}_{e,1} + \widetilde{V}_{e,2} = \widetilde{V}_e'$$

Step 2: Key Transformation for $u \leftarrow 1$ **to** 2 **do**
  for $v \leftarrow 1$ **to** 2 **do**
    for $w \leftarrow 1$ **to** 2 **do**
      $\widehat{EV}_e^{u,v,w} \leftarrow (\widehat{M}_L^{u,v,w})^{-1} \widehat{V}_{e,v}' \widehat{M}_R^{u,v,w}$
      $\widetilde{EV}_e^{u,v,w} \leftarrow (\widetilde{M}_L^{u,v,w})^{-1} \widetilde{V}_{e,v}' \widetilde{M}_R^{u,v,w}$
    end
  end
end

Step 3: Matrix Construction

$$EV_e \leftarrow \left\{ \widehat{EV}_e^{u,v,w}, \widetilde{EV}_e^{u,v,w} \;\middle|\; u,v,w \in \{1,2\} \right\}$$

return EV_e

Definition 13. Encrypted TSSD-matrix of Query Exemplar Dataset. Given the exemplar dataset D_e's $TV_e = \{\widehat{V_e'}, \widetilde{V_e'}\}$, the encrypted TSSD-matrix of TV_e is denoted as EV_e, each EV_e is generated by Eq.(6).

$$EV_e = Enc_e(TV_e, sk) = \left\{ \begin{matrix} (\widehat{M}_L^{u,v,w})^{-1} \widehat{V}_{e,v}' \widehat{M}_R^{u,v,w}, \\ (\widetilde{M}_L^{u,v,w})^{-1} \widetilde{V}_{e,v}' \widetilde{M}_R^{u,v,w} \end{matrix} \;\middle|\; u,v,w \in \{1,2\} \right\} \tag{6}$$

where Enc_e is the encryption Algorithm 2.

Encrypted vectors EV_i and matrix EV_e enable secure similarity computations while preserving confidentiality.

5.2 Privacy-preserving Similarity Comparison

After encryption, we implement the spatial similarity measure by a method that incorporates cryptographic similarity computation and comparison, defined as follows.

Definition 14. Encrypted Similarity Comparison. Given two spatial datasets D_i and D_j, along with an exemplar dataset D_e, we represent their encrypted forms as their TSD-vectors EV_i, EV_j and TSSD-matrix EV_e, respectively. The encrypted similarity between these datasets can be computed and compared using the following equations:

$$\begin{cases} \widehat{z}_{i,e,j} = \sum_{u,v,w=1}^2 \widehat{EV}_{i,L}^{u,v,w} \cdot \widehat{EV}_e^{u,v,w} \cdot \widehat{EV}_{j,R}^{u,v,w} \\ \widetilde{z}_{i,e,j} = \sum_{u,v,w=1}^2 \widetilde{EV}_{i,L}^{u,v,w} \cdot \widetilde{EV}_e^{u,v,w} \cdot \widetilde{EV}_{j,R}^{u,v,w}. \end{cases} \tag{7}$$

And then Compute $z_{i,e,j} = \widehat{z}_{i,e,j} + \widetilde{z}_{i,e,j}$. if $z_{i,e,j} > 0$, it shows that dataset D_i is more similar to exemplar dataset D_e than D_j.

Next, we present a theorem demonstrating that the relative similarity between spatial datasets can be determined through encrypted vector-matrix multiplication between TSD-vectors and TSSD-matrices.

Theorem 1. Given a exemplar dataset D_e, for any two spatial datasets D_i and D_j, if $z_{i,e,j} = \widehat{z}_{i,e,j} + \widetilde{z}_{i,e,j} > 0$, then $Sim(G_i, G_e) > Sim(G_j, G_e)$ holds.

Proof: According to Definition 10, Definition 11, and Definition 14, we derive the following expression for $\widehat{z}_{i,e,j}$:

$$\begin{aligned}\widehat{z}_{i,e,j} &= \sum_{u,v,w=1}^{2} \widehat{EV}_{i,L}^{u,v,w} \cdot \widehat{EV}_{e}^{u,v,w} \cdot \widehat{EV}_{j,R}^{u,v,w} \\ &= \widehat{\alpha}_{V_{i,L}} \cdot \widehat{\alpha}_{V'_e} \cdot \widehat{\alpha}_{V_{j,R}} \cdot V_{i,L}^T V'_e V_{j,R} + \widehat{\beta}_{V_{i,L}} \cdot \widehat{\beta}_{V'_e} \cdot \widehat{\beta}_{V_{j,R}} + \widehat{\gamma}_{V'_e}\end{aligned}$$

Similarly, for $\widetilde{z}_{i,e,j}$, we have:

$$\begin{aligned}\widetilde{z}_{i,e,j} &= \sum_{u,v,w=1}^{2} \widetilde{EV}_{i,L}^{u,v,w} \cdot \widetilde{EV}_{e}^{u,v,w} \cdot \widetilde{EV}_{j,R}^{u,v,w} \\ &= \widetilde{\alpha}_{V_{i,L}} \cdot \widetilde{\alpha}_{V'_e} \cdot \widetilde{\alpha}_{V_{j,R}} \cdot V_{i,L}^T V'_e V_{j,R} + \widetilde{\beta}_{V_{i,L}} \cdot \widetilde{\beta}_{V'_e} \cdot \widetilde{\beta}_{V_{j,R}} + \widetilde{\gamma}_{V'_e}\end{aligned}$$

Given that $\widehat{\gamma}_{V'_e} + \widetilde{\gamma}_{V'_e} = 0$, we can combine these to find $z_{i,e,j}$:

$$\begin{aligned}z_{i,e,j} &= \widetilde{z}_{i,e,j} + \widehat{z}_{i,e,j} \\ &= (\widehat{\alpha}_{V_{i,L}} \cdot \widehat{\alpha}_{V'_e} \cdot \widehat{\alpha}_{V_{j,R}} + \widetilde{\alpha}_{V_{i,L}} \cdot \widetilde{\alpha}_{V'_e} \cdot \widetilde{\alpha}_{V_{j,R}}) \cdot V_{i,L}^T V'_e V_{j,R} \\ &\quad + (\widehat{\beta}_{V_{i,L}} \cdot \widehat{\beta}_{V'_e} \cdot \widehat{\beta}_{V_{j,R}} + \widetilde{\beta}_{V_{i,L}} \cdot \widetilde{\beta}_{V'_e} \cdot \widetilde{\beta}_{V_{j,R}})\end{aligned}$$

Due to the random number settings in Definition 6, it follows that:

$$\widehat{\alpha}_{V_{i,L}} \cdot \widehat{\alpha}_{V'_e} \cdot \widehat{\alpha}_{V_{j,R}} + \widetilde{\alpha}_{V_{i,L}} \cdot \widetilde{\alpha}_{V'_e} \cdot \widetilde{\alpha}_{V_{j,R}} > \widehat{\beta}_{V_{i,L}} \cdot \widehat{\beta}_{V'_e} \cdot \widehat{\beta}_{V_{j,R}} + \widetilde{\beta}_{V_{i,L}} \cdot \widetilde{\beta}_{V'_e} \cdot \widetilde{\beta}_{V_{j,R}}$$

Thus, we can conclude that if $z_{i,e,j} > 0$, then $V_{i,L}^T \cdot V'_e \cdot V_{j,R} > 0$. According to Lemma 1, we can derive if $z_{i,e,j} = \widetilde{z}_{i,e,j} + \widehat{z}_{i,e,j} > 0$, then $Sim(G_i, G_e) > Sim(G_j, G_e)$ It's clear that Theorem 1 holds. ■

Theorem 1 indicates that the similarity relationship of spatial dataset D_i and D_j can be determined by whether the product of the encrypted vectors and matrix $z_{i,e,j}$ is greater than 0. Therefore, the top-k datasets with the highest similarity to the exemplar dataset can be determined without knowing any plaintext information of these datasets.

5.3 AME-Based Search Processing

In this section, we outline the Priekds process, which comprises two modules: the setup module and the search module. The setup module enables the data owner (DO) to preprocess the spatial datasets and subsequently outsource the encrypted data to the cloud server (CS). The search module, on the other hand, facilitates the exemplar dataset search processing between the data user (DU) and the CS.

- **Algorithms in the Setup Module**

SK $\leftarrow$ ***GenKey***(1^{ζ}). This algorithm is performed by DO to generate private keys. Taking the security parameter ζ as input, a set of keys $SK = \{sk, \theta\}$is generated, where θ is the grid partition threshold, and sk are the keys to encrypt TSD-vectors and TSSD-matrices. At the same time, the secret key sk will be granted to users u who have already registered in the system.

$G \leftarrow$ ***GenDenDist***$(\mathcal{D}, \theta)$.This algorithm is executed by the data owner (DO) to transform spatial datasets $\mathcal{D} = \{D_1, D_2, \cdots, D_n\}$ into density distributions $G = \{G_1, G_2, \cdots, G_n\}$ (Definition 4).

$\mathcal{EV} \leftarrow$ ***Encrypt***(G, SK). This algorithm is performed by DO to encrypt the spatial datasets and their density distributions, the implementation steps are as follows.

For each density distribution $G_i \in G$, G_i is first vectorised to a SD-vector V_i according to Definition 6, subsequently, V_i is augmented with random numbers, and then split into TSD-vector TV_i according to Definition 11, and last TV_i is encrypted to EV_i according to Definition 12. As a result, the set of encrypted TSD-vectors $\mathcal{EV}$ is generated.

Table 1: Details of spatial data repositories

Data repository	Storage (GB)	Number of datasets
Identifiable	19.64	235,483
Trackable	4.48	66,380

After encryption, $\mathcal{EV}$ are outsourced to CS, and the setup module is completed.

Algorithm 3 Secure Top-k Search

Input: Encrypted TSD-vectors set $\mathcal{EV}$, trapdoor $TD = (EV_e, k)$
Output: Search results R

Initialize max-heap H with capacity k Initialize result set $R \leftarrow \emptyset$
for *each* $EV_i \in \mathcal{EV}$ **do**
  if $|H| < k$ **then**
    H.insert(EV_i)
  else
    $EV_{\text{root}} \leftarrow H$.get-max() **if** $\textsc{AmeEval}(EV_i, EV_{root}, EV_e) = 1$ **then**
      Remove EV_{root} from H;
      Add EV_i to H;
    end
  end
end
Function $\textsc{AmeEval}(EV_i, EV_j, EV_e)$
  Compute $z_{i,e,j} \leftarrow \widetilde{z}_{i,e,j} + \widehat{z}_{i,e,j}$ **return** $(\, z_{i,e,j} > 0\,)$? 1 : 0
for *each* $EV_i \in H$ **do**
  R.add(EV_i);
end
return R;

- **Algorithms in the Search Module**

$TD \leftarrow$ ***GenTrapdoor***(D_e, SK, k). DU encrypts the exemplar dataset D_e into a search trapdoor $TD = \{EV_e, k\}$ by using the keys of SK (Definition **??**), where EV_e is the encrypted TSSD-matrix of D_e. And then, DU transmits the trapdoor TD to CS as the search command.

$R \leftarrow$ ***Search***(TD, EV). Once receiving a trapdoor from DU, CS performs the *search* algorithm (Algorithm 3) and returns the k most similar spatial datasets to D_e as the search result.

The *Search* algorithm searches for the k most similar spatial datasets to the exemplar dataset according to Theorem 1. The time complexity of *Search* is $O(n(d^2 + d^2 \log_2 k) + d^3)$, where d is the length of SD-vectors, d^2 represents the computational costs required for encrypting n spatial datasets during Local Data Outsourcing, The computational costs incurred when inserting the encrypted dataset into the max-heap for secure comparison are represented by $d^2 \log_2 k$. The computational expenditure required for encrypting the sample dataset is denoted by d^3.

6 Performance Evaluation

In this section, we first describe the experimental setup. Next, we thoroughly evaluate the proposed Priekds and Priekds+ schemes using three key metrics: search effectiveness, search efficiency, and space cost.

Table 2: Parameter settings

Notations	Meanings	Default values
k	the size of search result	10
n	the number of spatial datasets	15000
θ	the grid threshold	900
p	Number of chunks for vector partitioning	2
c	Number of clusters	20

6.1 Settings

Datasets. The proposed schemes are evaluated on three real-world spatial data repositories, *Identifiable*, *Public* and *Trackable*, which are used in paper [29]. The details and the spatial overhead in the proposed schemes of each data repository are shown in Table 1.

Implementation. The experimental hardware environment is Intel i7-14650HX CPU, 128GB memory and 1TB hard disk; the software environment is 64bit Windows 11 and Python 3.11 develop environment.

Exemplar datasets and parameter settings. The results are averaged over 100 searches, with each search using a randomly selected dataset from the repository as the exemplar. Parameter settings are detailed in Table 2.

6.2 Search Effectiveness Evaluation

In this section, we assess the performance of the Priekds and algorithms in the context of spatial dataset retrieval tasks. The MSE is utilized as a quantitative measure to determine the accuracy of the search outcomes.

To calculate the MSE, we first perform k-means clustering on the points in the exemplar dataset to determine the cluster centers. Subsequently, we compute the average distance from each point in the search results to its nearest cluster center. The MSE quantifies the average distance between the location points of the search results and those of the exemplar dataset, with a smaller MSE indicating closer proximity between the two sets of points. Figure 3 illustrate the MSE of Priekds and PriDAS as the parameters k, θ and vary.

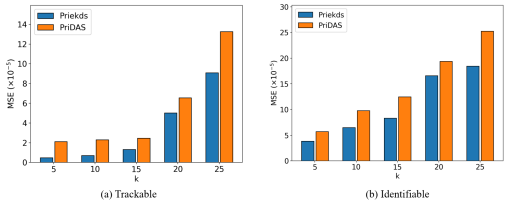

Figure 3: MSE versus k.

Figure 3 compares the MSE versus k for the Priekds and PriDAS algorithms. The parameter k indicates the number of spatial datasets requested by the data user. As k increases, the MSE of Priekds and PriDAS algorithms increases. This is because the larger k is, the more datasets are included in the search results and the more datasets with lower similarity, which leads to higher MSE values. Figure 4 compares MSE versus θ for the Priekds and PriDAS algorithms. The parameter θ denotes the space-dividing threshold used in the grid partitioning process. The parameter θ directly determines the number of grids within each dataset's DDSM, thereby

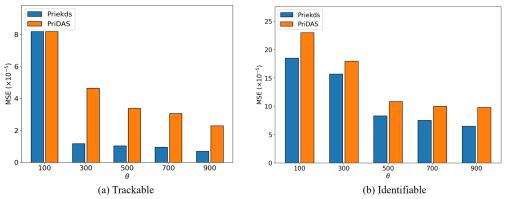

Figure 4: MSE versus θ.

affecting the computational time overhead for each similarity calculation. Figure 5 compares MSE versus c for the Priekds and PriDAS

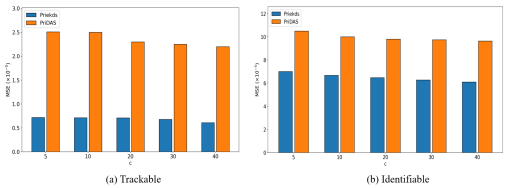

Figure 5: MSE versus c.

algorithms. The parameter c denotes the number of cluster centers used in clustering the search results. An increase in c implies a larger number of cluster centers, resulting in the dataset being partitioned into more categories. As c increases, the MSE values for Priekds and PriDAS all decrease. This decrease occurs because a higher number of cluster centers reduces the average distance between the search result points and their nearest cluster center.

In conclusion, our proposed Priekds achieves consistently low and stable MSE values, and comparisons with other algorithms demonstrate their effectiveness.

References

[1] Marco D. Adelfio, Sarana Nutanong, and Hanan Samet. 2011. Similarity search on a large collection of point sets. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Chicago, Illinois) *(GIS '11)*. Association for Computing Machinery, New York, NY, USA, 132–141. https://doi.org/10.1145/2093973.2093992

[2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. 2010. A view of cloud computing. *Commun. ACM* 53, 4 (2010), 50–58. https://doi.org/10.1145/1721654.1721672

[3] Arturs Backurs, Yihe Dong, Piotr Indyk, Ilya Razenshteyn, and Tal Wagner. 2020. Scalable nearest neighbor search for optimal transport. In *ICML*. 497–506.

[4] Alex Bogatu, Alvaro A. A. Fernandes, Norman W. Paton, and Nikolaos Konstantinou. 2020. Dataset Discovery in Data Lakes. In *ICDE*. 709–720.

[5] Sonia Castelo, Rémi Rampin, Aécio Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. 2021. Auctus: A dataset search engine for data discovery and augmentation. *Proceedings of the VLDB Endowment* 14, 12 (2021), 2791–2794.

[6] Chahil Choudhary, Narayan Vyas, and Umesh Kumar Lilhore. 2023. Cloud Security: Challenges and Strategies for Ensuring Data Protection. In *2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS)*. 669–673. https://doi.org/10.1109/ICTACS59847.2023.10390302

[7] Pedro Arthur de Fernandes Vasconcelos, Wensttay de Sousa Alencar, Victor Hugo da Silva Ribeiro, Natarajan Ferreira Rodrigues, and Fabio Gomes de Andrade. 2017. Enabling Spatial Queries in Open Government Data Portals. In *EGOVIS*, Vol. 10441. 64–79.

[8] Auriol Degbelo and Brhane Bahrishum Teka. 2019. Spatial search strategies for open government data: a systematic comparison. In *GIR*. 1–10.

[9] Marios D. Dikaiakos, Dimitrios Katsaros, Pankaj Mehra, George Pallis, and Athena Vakali. 2009. Cloud Computing: Distributed Internet Computing for IT and Scientific Research. *IEEE Internet Computing* 13, 5 (2009), 10–13. https://doi.org/10.1109/MIC.2009.103

[10] Patricia Frontiera, Ray Larson, and John Radke. 2008. A comparison of geometric approaches to assessing spatial similarity for GIR. *International Journal of Geographical Information Science* 22, 3 (2008), 337–360.

[11] Shabnam Kasra Kermanshahi, Shi-Feng Sun, Joseph K Liu, Ron Steinfeld, Surya Nepal, Wang Fat Lau, and Man Ho Allen Au. 2020. Geometric range search on encrypted data with forward/backward security. *IEEE Transactions on Dependable and Secure Computing* 19, 1 (2020), 698–716.

[12] David Kolevski, Katina Michael, Roba Abbas, and Mark Freeman. 2021. Cloud Data Breach Disclosures: the Consumer and their Personally Identifiable Information (PII)?. In *2021 IEEE Conference on Norbert Wiener in the 21st Century (21CW)*. 1–9. https://doi.org/10.1109/21CW48944.2021.9532579

[13] Pengyue Li, Hua Dai, Sheng Wang, Wenzhe Yang, and Geng Yang. 2024. Privacy-preserving Spatial Dataset Search in Cloud. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (Boise, ID, USA) *(CIKM '24)*. New York, NY, USA, 1245–1254.

[14] Luca Marconato, Giovanni Palla, Kevin A. Yamauchi, Isaac Virshup, Elyas Heidari, Tim Treis, Wouter-Michiel Vierdag, Marcella Toth, Sonja Stockhaus, Rahul B. Shrestha, Benjamin Rombaut, Lotte Pollaris, Laurens Lehner, Harald Vöhringer, Ilia Kats, Yvan Saeys, Sinem K. Saka, Wolfgang Huber, Moritz Gerstung, Josh Moore, Fabian J. Theis, and Oliver Stegle. 2025. SpatialData: an open and universal data framework for spatial omics. *Nat Methods* 22 (2025), 58–62. https://doi.org/10.1038/s41592-024-02212-x

[15] Yinbin Miao, Yutao Yang, Xinghua Li, Zhiquan Liu, Hongwei Li, Kim-Kwang Raymond Choo, and Robert H Deng. 2023. Efficient privacy-preserving spatial range query over outsourced encrypted data. *IEEE Transactions on Information Forensics and Security* 18 (2023), 3921–3933.

[16] Davide Mottin, Matteo Lissandrini, Yannis Velegrakis, and Themis Palpanas. 2014. Exemplar queries. *Proceedings of the VLDB Endowment* (Jan 2014), 365–376. https://doi.org/10.14778/2732269.2732273

[17] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas. 2016. Exemplar queries: a new way of searching. *The VLDB Journal* 25 (December 2016), 741–765. https://doi.org/10.1007/s00778-016-0429-2

[18] Sarana Nutanong, Edwin H Jacox, and Hanan Samet. 2011. An incremental Hausdorff distance calculation algorithm. *Proceedings of the VLDB Endowment* 4, 8 (2011), 506–517.

[19] Tran Phong, Shaw Rajib, Chantry Guillaume, and Norton John. 2008. GIS and local knowledge in disaster management: a case study of flood risk mapping in Viet Nam. *Disaster Prevention and Management* 17, 4 (2008), 498–510. https://doi.org/10.1111/j.1467-7717.2008.01067.x

[20] Jan Ramon and Maurice Bruynooghe. 2001. A polynomial time computable metric between point sets. *Acta Informatica* 37, 10 (2001), 765–780.

[21] Walter Renteria-Agualimpia, Francisco J Lopez-Pellicer, Javier Lacasta, F Javier Zarazaga-Soria, and Pedro R Muro-Medrano. 2016. Improving the geospatial consistency of digital libraries metadata. *Journal of Information Science* 42, 4 (2016), 507–523.

[22] Abdel Aziz Taha and Allan Hanbury. 2015. An efficient algorithm for calculating the exact Hausdorff distance. *IEEE transactions on pattern analysis and machine intelligence* 37, 11 (2015), 2153–2163.

[23] Qiuyun Tong, Yinbin Miao, Hongwei Li, Ximeng Liu, and Robert H Deng. 2021. Privacy-preserving ranked spatial keyword query in mobile cloud-assisted fog computing. *IEEE Transactions on Mobile Computing* 22, 6 (2021), 3604–3618.

[24] Sheng Wang, Zhifeng Bao, J. Shane Culpepper, Timos Sellis, Mark Sanderson, and Xiaolin Qin. 2017. Answering Top-k Exemplar Trajectory Queries. In *ICDE*. 597–608.

[25] Sheng Wang, Zhifeng Bao, Shixun Huang, and Rui Zhang. 2018. A Unified Processing Paradigm for Interactive Location-based Web Search. In *WSDM*. ACM, 601–609.

[26] Wai Kit Wong, David Wai-Lok Cheung, Ben Kao, and Nikos Mamoulis. 2009. Secure kNN computation on encrypted databases. In *SIGMOD*. 139–152.

[27] Guowen Xu, Hongwei Li, Yuanshun Dai, Kan Yang, and Xiaodong Lin. 2018. Enabling efficient and geometric range query with access control over encrypted spatial data. *IEEE Transactions on Information Forensics and Security* 14, 4 (2018), 870–885.

[28] Wenzhe Yang, Sheng Wang, Yuan Sun, Zhiyu Chen, and Zhiyong Peng. 2023. Efficient Spatial Dataset Search over Multiple Data Sources. *arXiv preprint arXiv:2311.13383* (2023).

[29] Wenzhe Yang, Sheng Wang, Yuan Sun, and Zhiyong Peng. 2022. Fast dataset search with earth mover's distance. *Proceedings of the VLDB Endowment* 15, 11 (2022), 2517–2529.

[30] Yandong Zheng, Rongxing Lu, Songnian Zhang, Jun Shao, and Hui Zhu. 2024. Achieving Practical and Privacy-Preserving kNN Query Over Encrypted Data. *IEEE Transactions on Dependable and Secure Computing* 21, 6 (2024), 5479–5492. https://doi.org/10.1109/TDSC.2024.3376084