Verifiable Learned PR-Tree Indexing for Privacy-Preserving Range Queries Over Encrypted Geospatial Data

Anagha Aher[1], Sangita Chaudhari[2]
Dept. of Computer Engineering, Ramrao Adik Institute of Technology, D. Y. Patil Deemed to be University, Nerul, India[1]
Dept. of Computer Science & Engineering, Ramrao Adik Institute of Technology,
D. Y. Patil Deemed to be University, Nerul, India[2]

***Abstract*—Protecting the privacy of geospatial data, along with efficient encrypted query processing, remains a major challenge in cloud-based GIS applications and location-based applications (LBS). In this study, a privacy-preserving framework for secure range query processing over encrypted vector geospatial data using a learned PR-tree index is integrated with an XGBoost-based bucket prediction model. In the first phase, the framework employs a lightweight dual-encryption scheme based on Lorentz and Galilean transformations. This encryption preserves coordinate relationships and enables reversible coordinate recovery. To improve query execution efficiency in the encrypted domain, the learned PR-tree predicts the most probable PR-tree buckets. This minimizes unnecessary search path traversal. Further, integrity verification during storage and query processing is ensured using the Merkle Hash Root and EdDSA digital signatures. Experimental evaluation was conducted using various real-world point, polyline, and polygon datasets. The proposed framework achieved prediction accuracy up to 97.2% with a very low mean bucket prediction error of 0.028. The obtained results demonstrate the efficiency and practicality of the proposed framework over encrypted cloud data.**

***Keywords*—Location privacy preservation; learned PR-tree indexing; range query processing; Merkle Hash Root verification; XGBoost for spatial prediction**

I. Introduction

Nowadays, the use of geospatial data for applications such as GIS services, urban planning, environmental monitoring, navigation, and emergency response systems has increased significantly [1]. This increased utilization of geospatial data has raised significant concerns about the security and privacy of geospatial information. Due to the large volume, geospatial data is stored in the cloud, leveraging the scalability, accessibility, and computational power of cloud service providers [2]. However, outsourcing geospatial data to semi-trusted cloud providers poses risks of unauthorized access, data tampering, and privacy breaches [3]. Here, the major challenge is maintaining the confidentiality and integrity of geospatial data while preserving its usability. Furthermore, efficient query execution over encrypted geospatial data is expensive due to excessive spatial index traversal during query processing.

Preserving location privacy is the primary challenge in securing geospatial data management against semi-trusted cloud providers and potential attackers. Many encryption techniques are proposed for data storage and retrieval, but they do not support efficient query processing [4][5]. However, directly encrypting a complete shapefile that stores geospatial data incurs excessive computational overhead. Traditional spatial indexing techniques, such as R-trees and PR-trees, support efficient query processing for plaintext spatial data. However, their performance degrades significantly when queries are executed over encrypted geospatial data. The geospatial vector data consists of coordinate-based spatial features such as points, lines, and polygons. It is important to preserve coordinate relationships and topological integrity after applying security transformations [6][7].

To address these challenges, this research proposes a secure, linear, and lightweight encryption technique, along with an efficient query-processing mechanism, for outsourcing geospatial data. The proposed method applies a dual encryption technique. To improve query efficiency over encrypted data, a learned PR-tree index based on an XGBoost bucket prediction model is constructed on encrypted spatial objects. The learned prediction model reduces unnecessary node traversal by identifying the most probable PR-tree buckets corresponding to encrypted query regions. To preserve integrity, Merkle Hash Root (MHR)-based verification metadata is generated for the encrypted spatial index as well as for the query processing workflow. To ensure there is no tampering of data, this Merkle Hash Root value is digitally signed using the EdDSA algorithm [8].

The key contributions of this study are summarized as follows:

- A dual encryption scheme based on Lorentz and Galilean-based encryption is proposed for privacy-preserving geospatial data storage and reversible coordinate recovery.
- A learned PR-tree index based on an XGBoost bucket prediction model is developed to reduce search path traversal and candidate node exploration during range query processing in the encrypted domain.
- An integrity verification mechanism based on Merkle Hash Root and EdDSA digital signatures is incorporated to detect unauthorized modification of encrypted geospatial data and query results.

This study is organized as follows: The related work section discusses existing research related to privacy-preserving geospatial query processing and learned spatial indexing techniques. The remaining sections consist of the proposed system architecture, the mathematical models and algorithms, and the performance evaluation and security analysis. The study ends with a conclusion section.

II. Related Work

Location-based services (LBS) represent a major application domain for encrypted spatial query techniques. As users share location information to receive personalized services, it is crucial to protect their privacy. This data explosion raises significant privacy concerns, as location information can reveal sensitive details about individuals' behaviors and preferences. To address these concerns, various techniques have been developed. To position the proposed framework, prior research is reviewed from three perspectives. These include location privacy-preservation techniques, secure query processing for spatial data, and spatial indexing techniques.

A. Location Privacy Preservation Techniques

Various approaches have been proposed for preserving location privacy in GIS systems. Albouq et al. proposed a double-obfuscation mechanism for preserving location privacy in IoT [9]. However, this approach does not support secure spatial indexing for efficient query execution. Zhang et al. proposed differential privacy with k-anonymity to preserve user trajectory privacy [10]. But the method incurs additional computational overhead for trajectory anonymization. A geo-indistinguishability model for indoor location-based services was proposed by Min et al. [11] in 2023. However, the framework primarily focuses on privacy preservation and lacks efficient query-processing support. Haydari et al. developed a framework for protecting mobility trajectories. Here, the injected noise affects spatial precision [12]. A local differential privacy framework for protecting 3D spatial coordinates using Hilbert encoding and randomized perturbation is developed by Yan [13]. However, the perturbation process may reduce query accuracy. Qiu et al. proposed a geo-obfuscation mechanism that generates locally relevant fake locations to protect user privacy [14]. This framework failed to support query processing in the encrypted domain.

B. Secure Query Processing for Spatial Data

Due to the large volume of geospatial data in the cloud, secure query processing over encrypted geospatial data has also gained significant attention. Wang et al. introduced a secure spatial query framework based on hidden vector encryption for protecting spatial and textual information [15]. But the framework incurs high computational complexity for large datasets. Another framework was proposed based on a privacy-preserving keyword similarity search technique for encrypted spatial datasets [16]. However, the approach mainly focuses on keyword similarity rather than on efficient range query execution. Zhao et al. developed a secure Boolean range query scheme using Hilbert curve mapping and Quadtree indexing [17]. In this case, leakage may still occur during index traversal. Gong et al. proposed a geographic keyword Boolean range query framework for encrypted spatial data. Still, the method suffers from scalability limitations on large encrypted geospatial datasets [18].

C. Spatial Indexing Techniques

Spatial indexing is important for efficiently managing large-volume spatial datasets in applications. The framework for crowdsourcing applications based on a secure, dynamic tree-based indexing structure is an efficient approach. But this approach does not integrate learned prediction mechanisms for optimized traversal. Wang et al. developed an encrypted Quadtree-based indexing framework in 2021 [19]. However, conventional tree traversal still increases query processing overhead. Miao et al. introduced a Geohash-based secure indexing mechanism [20]. This framework was designed for encrypted spatial range queries. As an extension, Miao et al. [21] introduced a GR-tree indexing structure for encrypted spatial query execution. However, the framework lacks support for integrity verification. The comparative analysis of existing privacy-preserving spatial query frameworks is highlighted in Table I.

TABLE I. Existing Privacy-Preserving Spatial Query Frameworks.

Framework	Query Support	Indexing Support	Integrity Verification	Findings
Obfuscation and Differential Privacy-Based Methods [9]–[14]	Partial	✗	✗	Improved privacy preservation, but reduced query accuracy, and a lack of encrypted-domain query execution.
Encryption-Based Secure Query Frameworks [15], [16]	✓	Partial	✗	Supports secure query execution with increased computational overhead.
Secure Spatial Range Query Frameworks [17], [18]	✓	Tree-based	✗	Efficient range query processing, but vulnerable to leakage and scalability issues.
Secure Spatial Indexing Techniques [19]–[21]	✓	Quadtree / GR-tree / Geohash	✗	Enhances query efficiency but lacks integrity verification and learned traversal optimization.
Proposed Learned PR-tree Framework	✓	Learned PR-tree	✓	Provides efficient, integrity-preserving, and scalable encrypted spatial query processing.

To address these limitations, the proposed framework introduces a learned PR-tree index integrated with an XGBoost-based bucket prediction model. The main purpose of this model is to offer efficient range query processing in the encrypted domain. In addition, the framework incorporates Merkle Hash Root and EdDSA-based verification to ensure the integrity of encrypted geospatial data and query results.

III. METHODOLOGY

In this section, the proposed framework for privacy-preserving range query processing over encrypted geospatial data is discussed. This framework integrates a dual-encryption scheme, a learned PR-tree index based on XGBoost bucket prediction. And the last phase includes an integrity verification mechanism that uses a Merkle Hash Root and EdDSA digital signatures.

A. Problem Definition

The original vector geospatial dataset is denoted by $D = \{G_1, G_2, \ldots, G_n\}$. Each spatial object G_i consists of ordered coordinate pairs representing point, polyline, or polygon geometries. The data owner applies the proposed dual encryption scheme to generate the encrypted dataset $D^E = \{G_1", G_2", \ldots, G_n"\}$, which consists of Lorentz-based encryption followed by Galilean-based encryption. Further, the learned PR-index I^Lon the encrypted data is constructed. The corresponding Merkle Hash root, Hroot, is calculated by the owner before sending the data to the semi-trusted cloud server.

The EdDSA digital signature is used to verify the Merkle Hash Root, ensuring integrity during data storage and query processing. In the later phase, a user-defined range query $Q = (xmin, ymin, xmax, ymax)$ is encrypted as Q" before being submitted to the cloud by the trusted end user. The cloud processes Q"over D^E using I^L and retrieves the encrypted result set $G' = g^{1'}, g^{2'}, \ldots, g_m'$. The query results G′, along with the associated Merkle verification metadata, are returned to the trusted end user. The trusted user verifies the integrity of the received results using the verification metadata. Only after successful verification is decryption performed to recover the original query result set. The objective of the proposed framework is to ensure data confidentiality, query privacy, and verifiability of query result integrity.

B. Core System Architecture Components

The proposed system architecture comprises four major entities: the Key Generation Center, the Geospatial Data Owner, the Cloud Service Provider, and the Trusted End User. This architecture collectively enables secure range-query processing over encrypted geospatial data.

1) The Key Generation Center: The key generation center is one of the most important entities of the system. It is responsible for generating encryption keys and cryptographic signatures. All these key values are dynamic, as they are calculated based on the features of the input shapefile. Additionally, the key generation center generates public and private keys for the Edwards-curve Digital Signature Algorithm (EdDSA). This digital signature is further used while sending the signed Merkle Hash Root values between the various entities. The EdDSA algorithm is used, as it provides faster signing and verification. Also, the signatures generated are shorter compared to other digital signature algorithms.

2) The Geospatial Data Owner Side: The geospatial data owner encrypts the spatial coordinates before storing them on the cloud. At first, the coordinate data owner receives the required keys from the key generation center to encrypt the data. This encryption first introduces obfuscation using Lorentz-based encryption by Galilean-based encryption. This dual encryption ensures that a potential attacker cannot regenerate the original data without access to the correct decryption key values. After encryption, the encrypted geospatial data is first organized into a PR-tree. Minimum Bounding Rectangles (MBRs) for all encrypted spatial objects for PR-tree construction. The spatial descriptors extracted from the encrypted MBRs are then used to train an XGBoost-based prediction model. This model learns a mapping from encrypted spatial regions to their corresponding PR-tree buckets. The trained prediction model is integrated with the PR-tree to construct the proposed learned PR-tree index. The learned PR-tree index improves query efficiency by reducing search path traversal in the encrypted domain. In the next step, the Merkle Hash Root is computed over the entire learned PR-tree structure. The computed Merkle Hash Root is digitally signed using EdDSA. Further, the encrypted dataset, the learned PR-tree index, and the signed Merkle Hash Root are transmitted to the cloud service provider. Fig. 1 shows the steps performed by the geospatial data owner.

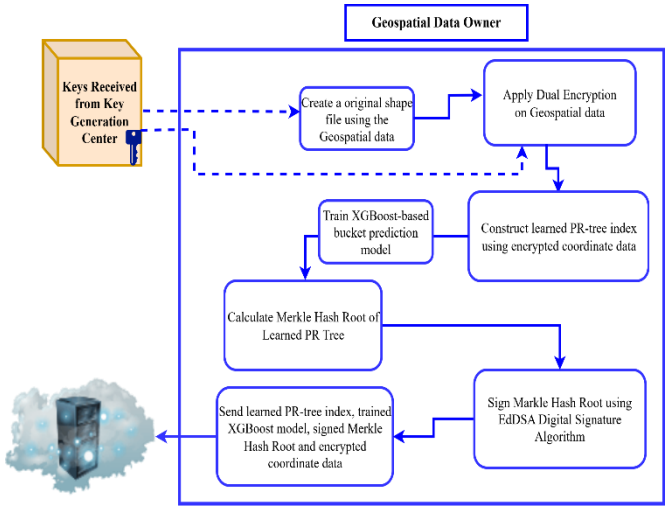

Fig. 1. The geospatial data owner.

3) The Cloud Service Provider Side: The cloud service provider is responsible for storing and processing the encrypted data. Initially, the cloud service provider receives and stores the learned PR-tree index along with the trained XGBoost prediction model. Along with the learned PR-tree index, the digitally signed Merkle Hash Root is also received. This Merkle Hash Root is used to verify the integrity of the data. The Merkle Hash Root of the reconstructed learned PR-tree is recalculated and compared with the Merkle Hash Root sent by the geospatial

data owner. Any unauthorized modification of the encrypted data or the learned PR-tree index can be detected through this verification process. When the trusted end user submits an encrypted query, the XGBoost prediction model first predicts the most probable PR-tree buckets. Later, the learned PR-tree index is traversed over the predicted regions to retrieve the encrypted query results. The Merkle Hash Root is generated for the retrieved query results. Then, this hash value is digitally signed using EdDSA and transmitted to the trusted end user along with the encrypted query results. The cloud service provider does not have access to the original geospatial coordinates. Fig. 2 shows the operational steps performed at the cloud service provider.

4) The Trusted End User: The trusted end user initiates range-query processing over the encrypted geospatial data stored by the cloud service provider. The query bounding box is encrypted using the proposed dual-encryption scheme before transmission. The query result received is also encrypted. The trusted end user verifies the integrity of the received query results by validating the received Merkle Hash Root verification metadata against the recomputed hash value. The cloud service provider verifies their integrity before decryption. In the last step, the received query results are decrypted to recover the original geospatial coordinates. Fig. 3 shows the processing steps performed by the trusted end user.

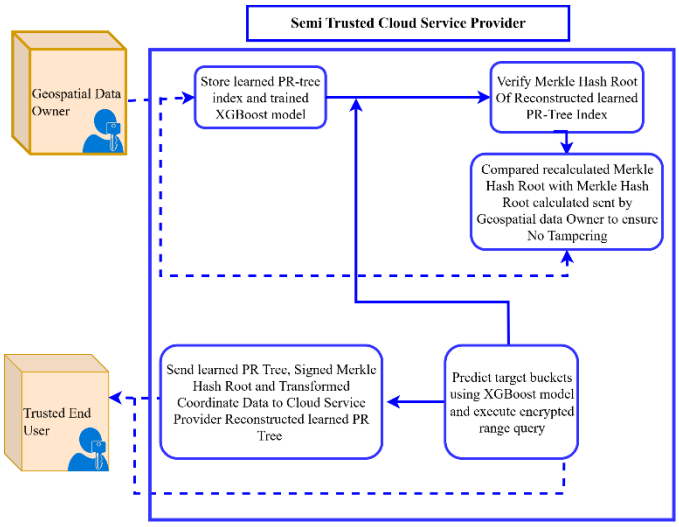

Fig. 2. The cloud service provider.

C. Proposed Dual Encryption Scheme

To ensure privacy preservation of geospatial vector data stored on semi-trusted cloud servers, this study proposes a dual-encryption model. The model combines Lorentz- and Galilean-based encryption to obfuscate spatial coordinates prior to storage. The encryption preserves coordinate relationships and enables accurate inverse reconstruction at the user side, thus ensuring location privacy, query efficiency, and shape preservation. The encryption parameters v, w, t, c, and γ are dynamically derived from the statistical features of the input shapefile. The proposed encryption scheme is reversible and preserves the coordinate relationships required for efficient spatial query processing. The proposed dual encryption and decryption processes are presented with mathematical equations as follows:

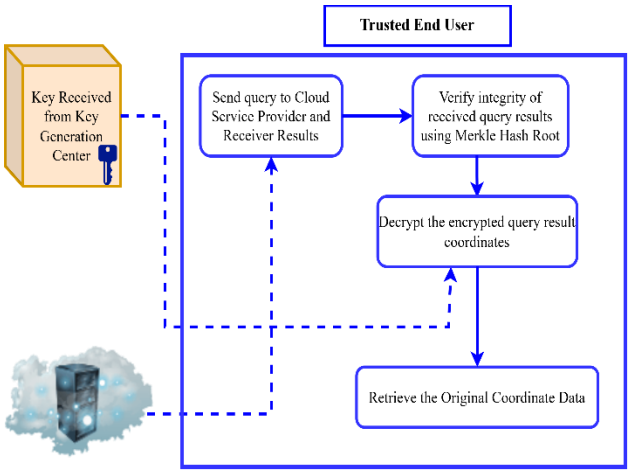

Fig. 3. The trusted end user.

Lorentz Factor Calculation:

$$\gamma = 1 / \sqrt{(1 - ((v^2 + w^2) / c^2))} \quad (1)$$

Lorentz-Based Encryption:

$$x' = \gamma(x - vt) \quad (2)$$

$$y' = \gamma(y - wt)$$

Galilean-Based Encryption:

$$x'' = x' + vt \quad (3)$$

$$y'' = y' + wt$$

Inverse Galilean Decryption:

$$x' = x'' - vt \quad (4)$$

$$y' = y'' - wt$$

Inverse Lorentz Decryption:

$$x = (x' / \gamma) + vt \quad (5)$$

$$y = (y' / \gamma) + wt$$

Finally, Algorithm 1 describes the complete flow of the encryption and decryption scheme.

Algorithm 1: Lorentz and Galilean-based Coordinate Encryption and Decryption

Input: Coordinate set $\{(x_i, y_i)\}$
Output: Original coordinate set $\{(x_i, y_i)\}$

Step 1: Derive dataset-dependent parameters
Compute v, w, t, c, and γ
Step 2: Perform Lorentz-based encryption
$x'_i = \gamma(x_i - vt)$
$y'_i = \gamma(y_i - wt)$
Step 3: Perform Galilean-based encryption
$x''_i = x'_i + vt$
$y''_i = y'_i + wt$
Step 4: Store encrypted coordinates $\{(x''_i, y''_i)\}$

Step 5: Perform inverse Galilean-based decryption
$x'_i = x''_i - vt$
$y'_i = y''_i - wt$
Step 6: Perform inverse Lorentz-based decryption
$x_i = (x'_i / \gamma) + vt$
$y_i = (y'_i / \gamma) + wt$

D. Learned PR Index Construction

The proposed learned PR-tree index is constructed over the encrypted coordinates to improve the efficiency of encrypted range query processing. Initially, Minimum Bounding Rectangles (MBRs) are generated for all encrypted spatial objects. Spatial descriptors extracted from the encrypted MBRs are further used for bucket-level learning. The proposed XGBoost-based prediction model is trained to learn the mapping between encrypted spatial regions and their corresponding PR-tree buckets. During query processing, the trained prediction model predicts the most probable target buckets. This reduces unnecessary traversal of unrelated PR-tree regions. The spatial descriptors extracted from the encrypted MBRs include cx (Centroid x-coordinate of encrypted MBR), cy (Centroid y-coordinate of encrypted MBR), w (width of encrypted MBR), h (height of encrypted MBR), and a (area of encrypted MBR). These descriptors are further used as input features for the XGBoost-based bucket prediction model as a feature vector $F = \{cx, cy, w, h, a\}$. The trained XGBoost prediction model estimates the corresponding PR-Tree bucket identifier for the encrypted spatial region as B = M(F). F represents the extracted encrypted MBR feature vector, M represents the trained XGBoost prediction model, B represents the predicted PR-Tree bucket identifier.

Algorithm 2 summarizes the workflow for constructing the proposed learned PR-tree index and training the XGBOOST model.

Algorithm 2: Learned PR-Tree Index Construction and Model Training

Input: Encrypted spatial objects $\{(x''_i, y''_i)\}$
Output: Learned PR-Tree Index

Step 1: Generate encrypted MBRs for all encrypted spatial objects.
Step 2: Construct the initial PR-Tree using encrypted MBRs.
Step 3: Extract spatial descriptors from encrypted MBRs:
- centroid coordinates (cx, cy)
- width (w)
- height (h)
- area (a)
Step 4: Assign corresponding PR-Tree bucket identifiers.
Step 5: Train XGBoost prediction model using:
Input Features → {cx, cy, w, h, a}
Target Label → bucket id
Step 6: Integrate the trained XGBoost model with the PR-Tree structure.
Step 7: Predict probable PR-Tree buckets during query processing.
Step 8: Search neighboring buckets if a prediction mismatch occurs.

E. Range Query Processing

Using the learned PR-tree index, the proposed range query processing framework enables efficient execution. Initially, the trusted end user encrypts the query bounding coordinates using the proposed dual encryption scheme.

The XGBoost-based bucket prediction model identifies the most likely bucket identifiers for the encrypted query region. Further, the learned PR-tree index traverses only the predicted and neighboring buckets to reduce unnecessary search operations. The integrity verification metadata, based on the Merkle Hash Root and EdDSA digital signatures, is used to ensure that the query results are not tampered with during processing or transmission. Finally, the trusted end user decrypts the encrypted query results to recover the original spatial coordinates. The encrypted query region is represented as $Q'' = E(Q)$. Q represents the original range query. E represents the proposed dual encryption function, and Q″ represents the encrypted range query. Finally, the encrypted query result generated using the learned PR-tree index is represented as $R' = S(Q'', I^L)$. R′ represents the encrypted query result. S represents the learned PR-tree query processing operation. I^L represents the learned PR-tree index.

IV. Results and Performance Evaluation

This section presents the experimental evaluation of the proposed framework to assess its effectiveness in query processing over encrypted data. The performance is evaluated in terms of accuracy, index construction cost, computational efficiency, and query execution across all geometries. This section evaluates the framework's ability to preserve spatial relationships and geometric structures while ensuring efficient range query processing.

A. Experimental Setup

For experimentation, six real-world datasets of point, line, and polygon layer geometries were considered in the EPSG:4326 coordinate reference system. To ensure spatial diversity, two datasets were selected for each geometry type. The proposed framework was implemented on the Google Collaboratory platform (Python 3 notebook environment). The system configuration consists of an Intel Xeon 2.2 GHz CPU with 12 GB of RAM. The proposed algorithms for the encryption scheme, hash-based indexing, and range query execution were applied to these datasets to validate the framework's correctness. Table II summarizes each dataset and its geospatial characteristics. Further, Table III highlights the experimental parameter settings used during the implementation and evaluation of the proposed learned PR-tree framework.

TABLE II. Metadata of Experimental Datasets

Dataset	Geometry	Number of Features	Purpose of Evaluation
Sydney	Point	5000	Evaluation of encrypted point range query processing
Berkeley	Point	3000	Scalability analysis for point geometries
Thane	Polyline	946	Evaluation of encrypted query execution over line geometries
Paris	Polyline	7195	Analysis of learned PR-tree traversal efficiency
Seoul	Polygon	715	Evaluation of encrypted polygon query processing

New York	Polygon	2597	Scalability and storage overhead analysis

TABLE III. EXPERIMENTAL PARAMETER SETTINGS TABLE

Parameter Name	Values Used in Experiment
Coordinate reference system	EPSG:4326
Bucket size	64
Total buckets	12, 15, 41, 47, 79, 113
Learning model	XGBoost
Prediction target	bucket_id
Bucket prediction error bound	±1 bucket
Query type	Range query
Number of test queries	200 per dataset
Integrity method	Merkle Hash Root with EdDSA signature

B. Performance Analysis

This section presents the performance evaluation of the proposed privacy-preserving and secure query-processing framework. This analysis first examines the cryptographic overhead introduced by the proposed dual-encryption scheme. Later, the efficiency of learned PR-tree index construction, XGBoost-based bucket prediction, and encrypted range query processing is evaluated using various datasets. Further, the proposed learned PR-tree framework is compared with the conventional PR-tree and SLRQ approaches to analyze improvements in query execution time and overall indexing efficiency. This analysis demonstrates the scalability and effectiveness of the proposed framework for secure geospatial query processing.

1) Cryptographic overhead analysis: The cryptographic overhead of the proposed dual-encryption framework across all point, polyline, and polygon datasets is thoroughly evaluated. As per the results demonstrated in Table IV, the proposed encryption and decryption operations incur relatively low computational overhead across all geometry types. The encryption and decryption times increase with the number of spatial coordinates present in the datasets. Further, as the proposed dual encryption scheme is lightweight, it is suitable for practical privacy-preserving geospatial query processing applications.

TABLE IV. CRYPTOGRAPHIC OVERHEAD OF THE PROPOSED DUAL ENCRYPTION.

Dataset	Layer Type	Encryption Time (s)	Decryption Time (s)	Total Cryptographic Overhead (s)
Sydney	Point	0.482	0.391	0.873
Berkeley	Point	0.536	0.447	0.983
Thane	Polyline	2.200	1.988	4.187
Paris	Polyline	2.487	2.143	4.630
Seoul	Polygon	0.641	1.312	1.953

2) Index construction and prediction performance: Table V presents the indexing and prediction performance of the proposed learned PR-tree framework. The results demonstrate that the learned PR-tree index can be constructed efficiently with relatively low index construction overhead. Also, it is observed that due to higher spatial complexity, the ML training time increases for large polyline datasets. However, the XGBoost-based bucket prediction model consistently achieves high accuracy across all datasets, reaching 97.2% for the Sydney point dataset. Fig. 4 demonstrates the efficiency of the proposed framework in terms of cryptographic overhead and learned indexing performance.

TABLE V. EVALUATION OF INDEX CONSTRUCTION AND PREDICTION ACCURACY.

Dataset	Layer Type	Index Construction Time (s)	ML Training Time (s)	Prediction Accuracy	Mean Bucket Prediction Error
Sydney	Point	0.184	2.918	0.972	0.028
Berkeley	Point	0.231	3.427	0.964	0.036
Thane	Polyline	0.582	17.296	0.875	0.125
Paris	Polyline	0.694	18.842	0.892	0.108
Seoul	Polygon	0.097	2.746	0.953	0.047
New York	Polygon	0.106	3.211	0.962	0.038

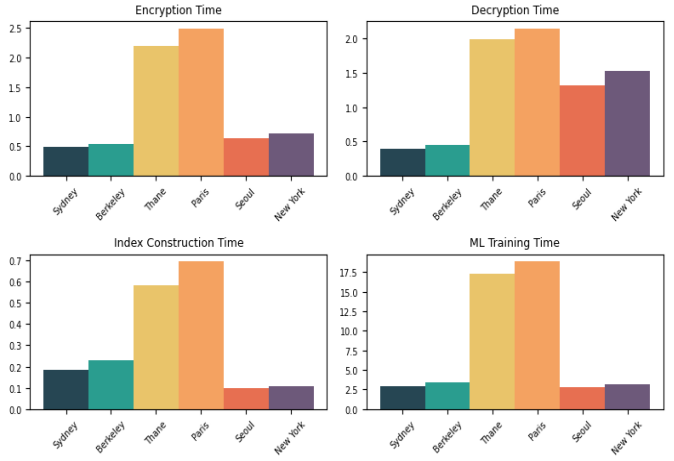

Fig. 4. Cryptographic overhead and learned PR-tree index construction performance across multiple geospatial datasets.

The lower prediction accuracy was observed for the polyline datasets (87.5% and 89.2%). This is primarily due to the higher geometric complexity of line features, including variations in length and orientation. The XGBoost model relies on MBR-based descriptors; geometrically distinct polylines may exhibit similar bounding-box characteristics, making bucket prediction more challenging. This affects only the efficiency of candidate selection and does not impact query correctness, as final results are validated through exact spatial intersection filtering.

3) Query processing performance of learned PR-tree index: The results of query processing demonstrate that the learned bucket prediction mechanism significantly reduces unnecessary traversal of unrelated PR-tree regions. During encrypted range query processing, the candidate reduction percentage remains consistently high across all datasets. This minimizes the number of candidate geometries considered during query execution. To assess the robustness of the proposed learned PR-tree framework, 200 randomly generated range queries were processed and analyzsed for each dataset. As shown in Table VI, the average query execution time ranged from 12.77 ms to 46.20 ms across the datasets of different geometry. Candidate reduction rates varied between 74.00% and 97.40%. This clearly indicates effective pruning of irrelevant spatial objects prior to final verification. Figure 5 demonstrates the efficiency of the proposed framework in terms of prediction accuracy, and encrypted query processing efficiency across different geospatial datasets.

TABLE VI. QUERY PROCESSING PERFORMANCE EVALUATION

Dataset	Layer Type	Avg. Query Time of 200 Queries (ms)	Std. Dev. (ms)	Candidate Reduction (%)
Sydney Point	Point	36.13	15.88	96.16
Berkeley Point	Point	17.30	10.71	93.77
Thane Polyline	Polyline	46.20	27.98	80.73
Paris Polyline	Polyline	30.97	25.26	97.40
Seoul Polygon	Polygon	12.77	11.81	74.00
New York Polygon	Polygon	22.41	14.10	92.77

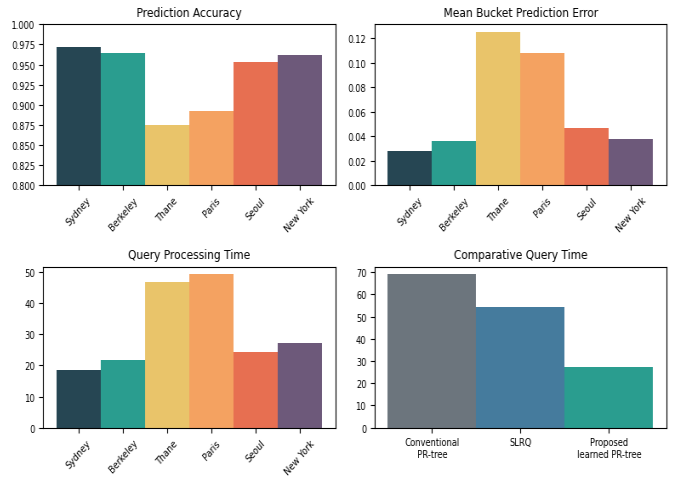

Fig. 5. Prediction accuracy and query processing performance analysis of the proposed learned PR-tree framework.

4) Comparative performance analysis with conventional PR-tree and SLRQ framework: The proposed learned PR-tree index framework significantly outperforms both the conventional PR-tree, SLRQ [17], and GRTree[21] approaches. During encrypted range query processing, the integration of the XGBoost-based bucket prediction model minimizes unnecessary search path traversal by directly identifying the most probable PR-tree buckets as shown in Figure 6 . As a result, the proposed framework achieves the lowest average query execution time and the highest candidate reduction percentage compared to the Conventional PR-tree, GRTree and the SLRQ framework, as presented in Table VII. The values in Table VII represent averages obtained across six datasets comprising point, polyline, and polygon geometries.

TABLE VII. COMPARATIVE PERFORMANCE ANALYSIS OF QUERY PROCESSING.

Method	Avg. Query Processing Time (ms)	Candidate Reduction (%)	Prediction-Based Traversal	ML-Based Optimization
Conventional PR-tree	68.91	61.34	No	No
SLRQ	54.26	69.47	No	No
GRTree	30.91	83.57	No	No
Proposed learned PR-tree	27.63	89.14	Yes	Yes

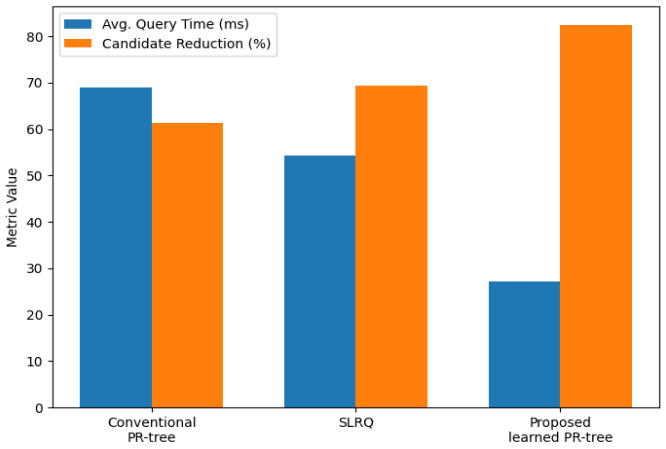

Fig. 6. Comparative performance with conventional PR-tree and SLRQ.

V. SECURITY ANALYSIS

To evaluate the security properties of the proposed privacy-preserving range query framework, a comprehensive security analysis was conducted under multiple attack scenarios. This includes key sensitivity analysis, query-forgery attacks, and index-poisoning attacks. The framework assumes that the cloud server is honest but curious. It has access only to transformed coordinates, encrypted MBR features, and the learned PR-tree index, while the transformation parameters and verification keys remain exclusively with the data owner.

A. Key Sensitivity Analysis

Controlled perturbations ranging from 0.1% to 10% were introduced to the decryption parameters to evaluate parameter sensitivity. The reconstructed coordinates were compared with the original coordinates using Mean Squared Error (MSE) and

Euclidean distance measures. As shown in Table VIII, the average reconstruction error increased from 0.097 units for a 0.1% perturbation to 9.063 units for a 10% perturbation. The MSE increased from 0.0047 to 41.0688. These results show that the attacker must have precise knowledge of the transformation parameters to recover coordinates accurately. Even minor deviations produce significant reconstruction errors.

TABLE VIII. KEY SENSITIVITY ANALYSIS

Key Perturbation (%)	MSE	Average Euclidean Error
0.1	0.0047	0.097
1.0	0.4609	0.960
5.0	10.9196	4.673
10.0	41.0688	9.063

B. Query Forgery Attack

A query forgery attack was simulated by modifying encrypted query results. During integrity validation, the Merkle Hash Root (MHR) signed with EdDSA is compared with the calculated Merkle Hash Root (MHR). The effectiveness of the attack was measured using the Attack Success Rate (ASR):

$$ASR@k = \frac{N_{correct}}{N_{attack}} \quad (6)$$

where, N undetected denotes the number of forged query results accepted by the verifier, and N forged represents the total forged attempts. For 250 forged query attempts per dataset, all modifications were successfully detected, resulting in an ASR of 0.0.

C. Index Poisoning Attack

An index poisoning attack was conducted by modifying encrypted objects stored in the learned PR-tree. The Tamper Detection Rate (TDR) was used to evaluate the effectiveness of the integrity verification mechanism:

$$TDR = \frac{N_{detected}}{N_{tampered}} \quad (7)$$

where, $N_{detected}$ represents the number of detected index modifications and $N_{tampered}$ denotes the total number of tampered index instances. For 250 poisoning attempts per dataset, all modifications were successfully detected, achieving a TDR of 1.0. Table IX presents the analysis of the tamper detection rate on different datasets.

TABLE IX. ANALYSIS OF THE TAMPER DETECTION RATE ON DIFFERENT DATASETS.

Dataset	Tampered Index Attempts	Detected	TDR
Sydney Points	250	250	1.000
Paris Roads	250	250	1.000
New York Parcels	250	250	1.000

The experimental results demonstrate that the proposed framework exhibits strong parameter sensitivity. It provides complete detection of query forgery and index poisoning attacks through the Merkle Hash Root and EdDSA-based verification mechanism. The proposed framework was also analyzed with respect to chosen-plaintext and statistical inference attacks. The cloud server has access only to encrypted coordinates, encrypted MBR features, and the learned PR-tree index. The encryption parameters remain exclusively with the data owner. Thus, direct recovery of the original coordinates is non-trivial. The framework is primarily designed for lightweight privacy-preserving query processing and verifiable computation. The proposed Lorentz and Galilean-based coordinate encryption effectively preserves the integrity of outsourced spatial data and query results.

VI. CONCLUSION

A privacy-preserving framework for secure range query processing over encrypted 2D vector geospatial data has been developed and evaluated in this work. This study presented a verifiable learned PR-tree indexing framework for privacy-preserving secure range query processing. The proposed framework integrates a lightweight dual encryption scheme based on Lorentz and Galilean transformations. Further, an XGBoost-based learned PR-tree index is constructed on encrypted coordinate data. The Merkle Hash Root verification mechanism with EdDSA digital signatures to ensure integrity preservation during cloud storage and query execution. The experimental results demonstrated that the proposed framework achieves efficient cryptographic performance with low encryption overhead. The learned PR-tree index achieved a high prediction accuracy of up to 97.2% and a low bucket prediction error. The proposed learned PR-tree indexing reduced average query processing time to 27.14 ms. It also achieved an 82.37% reduction in the number of candidates, compared to the conventional PR-tree and SLRQ frameworks. The obtained results confirm that the proposed framework provides an efficient, scalable, and secure solution for practical privacy-preserving geospatial query processing in semi-trusted cloud environments. The framework can be evaluated for complex queries and multi-dimensional geospatial data in the future.

REFERENCES

[1] A. Bhardwaj and V. Kumar, “Privacy and Healthcare during COVID-19,” in *Cybersecurity Crisis Management and Lessons learned from the COVID-19 Pandemic*. Hershey, PA, USA: IGI Global, 2022, pp. 82–96.

[2] L. Xiong, Z. Xu, and Y. Xu, “A secure re-encryption scheme for data services in a cloud computing environment,” *Concurrency Computat. Pract. Exper.*, vol. 27, no. 17, pp. 4573–4585, 2015.

[3] S. Ahmad, M. Arif, J. Ahmad, M. Nazim, and S. Mehfuz, “Convergent encryption enabled secure data deduplication algorithm for cloud environment,” *Concurrency Computat. Pract. Exper.*, vol. 36, no. 21, p. e8205, 2024.

[4] M. Milani, Y. Huang, and F. Chiang, “Data anonymization with diversity constraints,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3603–3618, 2021.

[5] B. Li and K. He, “Local generalization and bucketization technique for personalized privacy preservation,” *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 1, pp. 393–404, 2023.

[6] A. Majeed and S. O. Hwang, “Rectification of syntactic and semantic privacy mechanisms,” *IEEE Security Privacy*, vol. 21, no. 5, pp. 18–32, 2022.

[7] J. Wieringa, P. K. Kannan, X. Ma, T. Reutterer, H. Risselada, and B. Skiera, “Data analytics in a privacy-concerned world,” *J. Bus. Res.*, vol. 122, pp. 915–925, 2021.

[8] Y. Guan, R. Lu, Y. Zheng, S. Zhang, J. Shao, and G. Wei, “Toward privacy-preserving cybertwin-based spatiotemporal keyword query for ITS in 6G era,” *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16243–16255, 2021.

[9] S. S. Albouq, A. A. Abi Sen, A. Namoun, N. M. Bahbouh, and A. B. Alkhodre, "A double obfuscation approach for protecting the privacy of IoT location based applications," Scientific Reports, vol. 10, no. 1, Art. no. 7861, 2020, doi: 10.1038/s41598-020-64312-y.

[10] C. Zhang, L. Zhu, C. Xu, K. Sharif, X. Liu, and Y. Ding, "Protecting the moving user's locations by combining differential privacy and k-anonymity under temporal correlations in wireless networks," Wireless Communications and Mobile Computing, vol. 2021, Art. no. 6691975, 2021, doi: 10.1155/2021/6691975.

[11] M. Min, L. Xiao, J. Ding, H. Zhang, and S. Li, "3D geo-indistinguishability for indoor location-based services," IEEE Transactions on Mobile Computing, vol. 21, no. 11, pp. 4084–4097, 2022, doi: 10.1109/TMC.2021.3070192.

[12] A. Haydari, C.-N. Chuah, M. Zhang, J. Macfarlane, and S. Peisert, "Differentially private map matching for mobility trajectories," Proceedings on Privacy Enhancing Technologies, vol. 2022, no. 4, pp. 400–419, 2022, doi: 10.2478/popets-2022-0110.

[13] Y. Yan, P. Yan, A. Mahmood, Y. Zhang, and Q. Z. Sheng, "Achieving local differential location privacy protection in 3D space via Hilbert encoding and optimized random response," IEEE Transactions on Mobile Computing, vol. 23, no. 5, pp. 5234–5248, 2024, doi: 10.1109/TMC.2023.3298765.

[14] C. Qiu, R. Liu, P. Pappachan, A. Squicciarini, and X. Xie, "Time-efficient locally relevant geo-location privacy protection," IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 4, pp. 2891–2906, 2024, doi: 10.1109/TDSC.2023.3321456.

[15] X. Wang, J. Ma, F. Li, X. Liu, and Y. Miao, "Enabling efficient spatial keyword queries on encrypted data with strong security guarantees," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 4613–4628, 2021, doi: 10.1109/TIFS.2021.3106166.

[16] X. Song, C. Dong, D. Yuan, Q. Xu, and M. Zhao, "Privacy-preserving keyword similarity search over encrypted spatial data in cloud computing," IEEE Internet of Things Journal, vol. 9, no. 8, pp. 6184–6198, 2022.

[17] X. Zhao, J. Yu, X. Ge, and R. Hao, "Towards efficient secure boolean range query over encrypted spatial data," IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 2, pp. 1234–1248, 2023, doi: 10.1109/TDSC.2022.3156789.

[18] Z. Gong, J. Li, Y. Lin, J. Wei, and C. Lancine, "Efficient privacy-preserving geographic keyword boolean range query over encrypted spatial data," IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 5, pp. 4145–4159, 2023, doi: 10.1109/TDSC.2022.3220290.

[19] X. Wang, J. Ma, and X. Liu, "Enabling efficient and expressive spatial keyword queries on encrypted data," IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 6, pp. 2731–2745, 2021, doi: 10.1109/TDSC.2019.2963339.

[20] Y. Miao, C. Xu, Y. Zheng, X. Liu, and X. L. Meng, "Efficient and secure spatial range query over large-scale encrypted data," IEEE Transactions on Services Computing, vol. 16, no. 4, pp. 2789–2802, 2023, doi: 10.1109/TSC.2023.3264791.

[21] Y. Miao, Y. Yang, X. Li, L. Wei, and Z. Liu, "Efficient privacy-preserving spatial data query in cloud computing," Journal of Cloud Computing, vol. 12, no. 1, Art. no. 89, 2023, doi: 10.1186/s13677-023-00467-8.