GeoGraphJSON: A lightweight semantic data model integrating spatial geometry and graph connectivity for AI-Driven spatial reasoning

Muhamad Alrajhi[1], Christian Heipke[2], Mohammed Afroz[1]
[1]RASIKH Institute for Education and Training Riyadh, [2]Leibniz Universität Hannover

Keywords: GeoGraphJSON, GeoAI, Spatial Data Model, Graph Connectivity, Urban Analytics, Digital Twin.

Abstract

This paper presents GeoGraphJSON, a lightweight semantic data model that integrates spatial geometry with graph-based connectivity to support advanced spatial analysis and GeoAI applications. Conventional geospatial formats primarily focus on geometry and attributes, requiring relationships such as adjacency, containment, and connectivity to be derived dynamically through computationally intensive spatial operations. In contrast, GeoGraphJSON encodes these relationships explicitly as typed edges within a unified, interoperable structure. The proposed model introduces a hierarchical Unique Identifier (UID) system to ensure consistent lineage and cross-layer linkage, along with a structured edge framework to represent containment, adjacency, functional, and administrative relationships. A validation-driven methodology is developed to transform multi-layer geospatial datasets into graph-enhanced representations through data standardization, UID assignment, and rule-based edge generation.

The framework is implemented using a large-scale urban dataset from Riyadh comprising approximately 10,859 nodes and 13,733 edges across administrative, transportation, and urban asset layers. Graph-based evaluation demonstrates realistic spatial patterns, including right-skewed degree distribution, strong network connectivity, and identifiable community structures. Comparative analysis shows that GeoGraphJSON effectively bridges the gap between geometry-centric GIS formats and graph-based models, enabling efficient querying and direct integration with graph analytics workflows. The results highlight the model’s scalability and potential for supporting urban analytics, digital twins, and AI-driven geospatial systems.

1. Introduction

1.1 Motivation: From Geometry to Connectivity

Urban systems and infrastructure networks are inherently spatial, where nodes and edges are embedded within geographic space and influenced by physical distance and spatial constraints. Traditional graph-based models capture topological relationships but often fail to incorporate spatial properties, while geospatial models emphasize geometry without explicitly representing connectivity. This separation limits the ability to perform integrated spatial reasoning, particularly in AI-driven applications.

Spatial network theory highlights that real-world systems such as transportation, utilities, and urban environments are shaped by both topology and spatial embedding, where the cost of edge length and geographic proximity significantly influence network structure and connectivity patterns (Barthelemy, 2011).

GeoJSON provides an efficient and widely adopted representation of spatial features through geometries and feature collections (Butler et al., 2016); however, the specification does not define standardized mechanisms for explicitly encoding relationships or topological structures between features. To address this gap, this paper introduces GeoGraphJSON, a lightweight semantic data model that integrates spatial geometry with graph-based connectivity. The proposed approach enables unified representation of nodes, edges, and spatial attributes, supporting advanced analytics and AI-driven spatial reasoning.

1.2 Limitations of Existing Spatial Data Formats

Widely adopted geospatial formats including GeoJSON, Shapefile, Geodatabase (GDB), and GeoPackage (GPKG) are primarily designed for geometric representation and attribute storage. Although these formats support spatial analysis within GIS environments, they lack standardized mechanisms to persistently encode relationships between features. Topological relationships, such as containment and adjacency, are typically computed dynamically using overlay or proximity operations, rather than being stored as reusable connections. Similarly, hierarchical linkages across administrative levels (e.g., city → district → parcel → building) are often maintained through attribute-based references without consistent enforcement of lineage or validation rules.

More advanced models, such as CityGML and GeoSPARQL, introduce semantic capabilities and structured relationships; however, they often involve complex schemas and higher implementation overhead. Conversely, graph-based models provide strong support for representing relationships and network analysis but generally lack native support for spatial geometry and coordinate systems. This separation between spatial and relational representations results in fragmented workflows and reduced interoperability. These limitations highlight the absence of a unified, lightweight data model capable of integrating geometry, topology, and semantic relationships within a single interoperable structure.

1.3 Proposed Approach: GeoGraphJSON

To address these challenges, this paper introduces GeoGraphJSON, a lightweight semantic data model that integrates spatial geometry and graph connectivity within a unified framework. The model extends the GeoJSON structure by representing spatial entities as nodes and explicitly encoding relationships as typed edges, enabling both spatial and relational queries within the same dataset.

A key feature of GeoGraphJSON is the use of a hierarchical Unique Identifier (UID) system, which ensures consistent lineage, traceability, and cross-layer integration across multiple spatial levels. The framework defines standardized relationship types including containment, adjacency, functional, administrative, and semantic connections allowing complex

https://doi.org/10.5194/isprs-archives-XLIX-B4-2026-3-2026

urban interactions to be represented in a structured and machine-interpretable form. The model is designed to maintain compatibility with existing geospatial standards while enabling seamless integration with graph analytics platforms and GeoAI workflows. By bridging the gap between traditional GIS and graph-based systems, GeoGraphJSON provides a unified representation for spatial data and relationships.

1.4 Research Contributions

This paper presents the following contributions: a unified spatial–graph data model (GeoGraphJSON), a hierarchical UID-based indexing system, a typed edge framework for relationship modeling, a validation-driven data construction methodology, an implementation and evaluation using a real-world urban dataset, and a comparative assessment with existing models.

1.5 Paper Organization

The remainder of this paper is structured as follows:
Section 2 reviews related work on spatial and graph-based models. Section 3 presents the GeoGraphJSON data model. Section 4 describes the implementation methodology. Section 5 outlines the evaluation framework. Section 6 provides comparative analysis. Section 7 presents results and analytical insights. Section 8 discusses limitations and scalability, and Section 9 concludes the paper.

2. Related Work and Background

2.1 Spatial Networks

Spatial networks describe systems in which nodes and edges are embedded in geographic space, and connectivity is influenced by spatial constraints such as distance and cost. In such networks, the probability of connections is not uniform but depends on spatial proximity, leading to characteristic properties such as limited node degree, high clustering, and distance-dependent connectivity patterns (Barthelemy, 2011).

In urban and transportation systems, these constraints introduce trade-offs between efficiency and cost, where longer connections may improve accessibility but increase infrastructure complexity. This highlights the importance of integrating geometric and topological representations in spatial modeling.

2.2 Geometry-Centric Geospatial Data Models

Spatial data representation has evolved from traditional formats such as Shapefile and coverage datasets to modern web-oriented formats. These formats are primarily designed for geometric representation and visualization, offering limited support for semantic meaning or relational structure.

GeoJSON represents a significant advancement as a lightweight, human-readable, and web-compatible format for spatial data exchange (Butler et al., 2016). It supports representation of geometries such as points, lines, and polygons along with associated attributes and is widely used in web mapping and GIS applications. However, GeoJSON remains fundamentally geometry-centric, representing features as independent objects without explicit mechanisms to encode relationships such as adjacency, containment, or functional dependencies. Subsequent formats such as TopoJSON and CityJSON introduce improvements in efficiency and structural representation. TopoJSON reduces redundancy by preserving shared boundaries through explicit topology, while CityJSON provides a more compact and developer-friendly encoding of the CityGML data model, supporting semantic organization and hierarchical relationships through object-based linking (Ledoux et al., 2019). Despite these enhancements, these models do not provide a unified, lightweight framework for representing multi-layer spatial relationships in a graph-like structure.

2.3 Graph-Based Spatial Modeling

Graph-based data models represent entities as nodes and relationships as edges, enabling efficient modeling of connectivity, traversal, and relational querying. In geospatial contexts, place graph models encode places as nodes and spatial relationships as edges, supporting reasoning and knowledge extraction from spatial descriptions (Chen et al., 2018). Graph database platforms such as Neo4j facilitate efficient storage and traversal of such connected data structures, while analytical libraries such as NetworkX are widely used for network analysis in broader applications.

In practice, graph representations in geospatial applications are typically derived from existing GIS datasets by transforming spatial features into nodes and relationships such as adjacency, connectivity, or containment into edges. While this transformation enables powerful network-based analysis, it often results in duplication of data and a separation between the graph structure and the original geospatial representation. Furthermore, most graph-based models do not natively support geometric structures or coordinate reference systems, instead treating spatial information as attributes, which limits seamless integration with conventional GIS environments.

2.4 Semantic and Knowledge-Based Approaches

Recent developments have explored the integration of semantic web technologies and knowledge graphs with geospatial data. Standards such as GeoSPARQL enable spatial reasoning within RDF-based frameworks by supporting queries that combine geometric operations with semantic relationships (OGC, 2012). Similarly, approaches such as JSON-FG and Linked Building Data extend geospatial representations with metadata, ontologies, and semantic context. While these frameworks provide strong capabilities for semantic reasoning and interoperability, they often introduce additional complexity in data modeling, system implementation, and computational processing.

For instance, CityGML offers a comprehensive semantic model for 3D urban environments, yet its hierarchical structure and verbosity can limit efficiency in large-scale or real-time applications (Ledoux et al., 2019). Likewise, RDF-based approaches typically rely on specialized data infrastructures and query mechanisms, which may pose challenges for seamless integration within conventional GIS workflows (OGC, 2021; ISO, 2014).

2.5 Comparative Analysis of Existing Approaches

Geometry-centric formats provide simplicity and interoperability but lack explicit relationship modeling. Graph-based models enable efficient representation of connectivity but do not natively support spatial geometry. Semantic frameworks offer rich expressiveness but introduce complexity and implementation challenges.

Table 1 summarizes the key characteristics of widely used spatial and graph-based data models.

Model/Format	Geo.	Rel.	Sem.	Comp.
Shapefile	✓	✕	✕	L
GeoJSON	✓	✕	✕	L
TopoJSON	✓	◐	✕	M
CityGML	✓	◐	✓	H
GeoSPARQL	✓	✓	✓	VH
Graph Models	✕	✓	◐	M
GeoGraphJSON	✓	✓	✓	M

Table 1. Comparison of Spatial and Graph Data Models

Legend: Geo = Geometry Support; Rel. = Relationship Representation; Sem. = Semantic Support; Comp. = Complexity; ✓ Supported; ✕ Not Supported; ◐ Moderate; L = Low; M = Medium; H = High; VH = Very High

2.6 Research Gap and Positioning

The review highlights a clear gap in existing geospatial data models. No widely adopted approach provides a unified, lightweight structure that integrates geometry, topology, and semantic relationships while remaining compatible with standard GIS workflows. GeoGraphJSON addresses this gap by embedding graph connectivity and semantic relationships directly within a GeoJSON-compatible structure. By representing spatial entities as nodes and relationships as typed edges, the model enables both spatial and relational queries within a single dataset, supporting advanced analytics, validation, and AI-driven spatial reasoning.

3. GeoGraphJSON Concept and Data Model

3.1 Overview

GeoGraphJSON is a lightweight spatial data model that integrates geometric representation with graph-based connectivity and semantic relationships in a unified, interoperable structure. The model extends the GeoJSON standard by embedding explicit relationships between spatial entities, enabling both spatial and relational queries within a single dataset.

The core principle of GeoGraphJSON is that each spatial entity is represented as a node, and each relationship is explicitly defined as an edge. This approach enables urban assets to be modeled not only in terms of their geometry and attributes but also through their connections and functional relationships. As a result, the framework supports the simultaneous representation of spatial geometry (location and shape), topological relationships (connectivity and hierarchical structure), and semantic attributes (classification and function). By embedding graph structures within a geospatial framework, the model reflects key characteristics of spatial networks, where geographic constraints and connectivity jointly influence network structure and behavior (Barthelemy, 2011).

As a result, GeoGraphJSON transforms static spatial datasets into structured spatial graphs suitable for analytics, validation, and AI-driven reasoning.

The conceptual architecture of the GeoGraphJSON framework is illustrated in Figure 1.

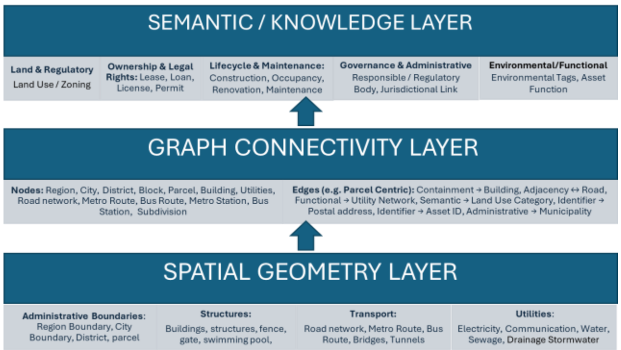

Figure 1. Conceptual architecture of GeoGraphJSON

3.2 Core Conceptual Model

The GeoGraphJSON model is based on three fundamental components: nodes, edges, and metadata.

3.2.1 Nodes (Spatial Entities): Each spatial feature is represented as a node that encapsulates geometry, attributes, and hierarchical references. Nodes may represent administrative units (e.g., region, city, district), land entities (parcels and buildings), transportation elements (road segments and nodes), or urban assets such as service points, street lights, and trees. Each node contains a unique identifier (feature_uid), geometry in GeoJSON format, attribute information such as land use and ownership, and a reference to its parent entity (parent_uid). This structure enables consistent identification and hierarchical linkage across multiple spatial layers.

3.2.2 Edges (Relationships): Relationships between spatial entities are represented as explicit, typed edges connecting nodes. Each edge includes a unique identifier (edge_uid), a relationship type (edge_type), a source node (source_uid), and a target node (target_uid). By explicitly storing relationships, GeoGraphJSON eliminates the need for repeated spatial computations (e.g., overlay or proximity analysis) and enables direct traversal and reasoning over spatial connectivity.

3.2.3 Metadata and Governance:
To support data governance and quality assurance, GeoGraphJSON incorporates metadata fields such as data source, update timestamp, responsible authority, confidence level, and spatial accuracy indicators. These attributes facilitate data validation, lifecycle tracking, and integration across systems. Figure 2 illustrates the conceptual structure of the GeoGraphJSON model.

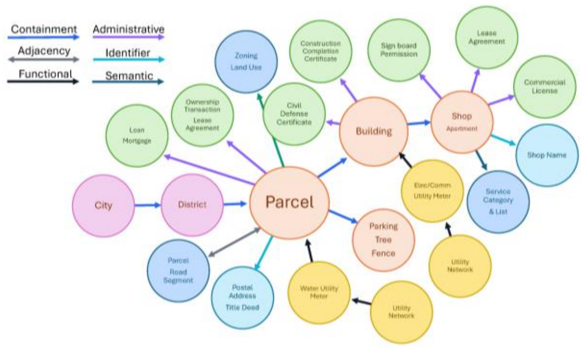

Figure 2. Conceptual GeoGraphJSON model

3.3 Hierarchical UID Framework

A key component of GeoGraphJSON is the hierarchical Unique Identifier (UID) system, which provides structured, machine-readable identification for all spatial entities. The UID encodes both entity type and hierarchical lineage, enabling direct traceability across spatial levels.

A typical UID follows the pattern XXX–YYY–0000(–0000), where XXX represents the entity type, YYY represents the parent reference, and 0000 represents a sequential identifier. This structure ensures hierarchical inheritance, consistency through standardized identifiers, traceability across layers, and interoperability for efficient data integration and graph construction. The UID framework serves as the backbone for linking nodes and generating relationships across multiple datasets.

3.4 Relationship (Edge) Framework

GeoGraphJSON defines relationship types including containment for hierarchical inclusion, adjacency for spatial proximity, boundary for shared geometric edges, functional for service linkages, administrative for governance relationships, and semantic associations for classification and identification.

Edges may be directed (e.g., service flow) or bidirectional (e.g., adjacency), depending on the relationship type. Cross-layer relationships are also supported, enabling connections between different thematic layers such as parcels, roads, and assets. This framework allows complex urban interactions to be explicitly modeled and queried within a unified structure.

3.5 Data Schema Structure

GeoGraphJSON organizes data into two primary collections. The node collection stores spatial entities with attributes including feature_uid, feature_type, geometry, associated attributes, and parent_uid. The edge collection stores relationships between entities, including edge_uid, edge_type, source_uid, target_uid, source_layer, and target_layer.

The schema is JSON-based, ensuring interoperability with web and GIS systems, while also enabling direct conversion to graph databases such as Neo4j or analytical frameworks like NetworkX. The structure is extensible, allowing domain-specific attributes to be incorporated as needed.

3.6 Design Principles

The GeoGraphJSON model is guided by principles including integration of geometry and connectivity, explicit relationship encoding, hierarchical consistency, interoperability, scalability, and AI readiness.

3.7 Summary

GeoGraphJSON provides a unified spatial–graph data model that bridges traditional geospatial representation and graph-based reasoning. By integrating geometry, hierarchical structure, and explicit relationships, the model enables efficient querying, validation, and analytical processing within a single framework. This foundation supports the implementation and evaluation presented in the subsequent sections.

4. Methodology and Implementation Workflow

4.1 Overview

The GeoGraphJSON framework is implemented through a structured workflow that transforms conventional geospatial datasets into a unified spatial–graph representation. The methodology integrates data preparation, hierarchical structuring, graph construction, validation, and analytical processing within a reproducible pipeline. The approach is designed to ensure consistency, scalability, and interoperability with existing GIS and graph-based systems, while embedding validation mechanisms to maintain data integrity. The overall implementation workflow is shown in Figure 3.

Figure 3. GeoGraphJSON Implementation Workflow

4.2 Study Area and Dataset

A study dataset was developed for a district in Riyadh, incorporating multiple spatial layers that reflect administrative, transportation, and urban asset structures. The dataset includes administrative hierarchies, road networks, transport layers, and urban assets. The dataset was structured to represent multi-layer spatial relationships required for graph construction and validation, providing a realistic test environment for the proposed model. The study area and integrated urban layers are shown in Figure 4.

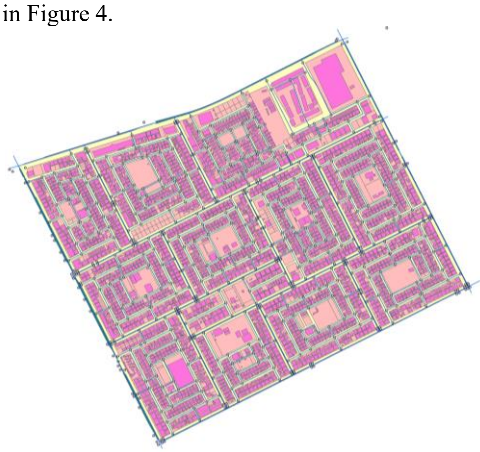

Figure 4. Study Area with Integrated Urban Layers

4.3 Data Preparation and Standardization

All spatial layers were standardized prior to graph construction to ensure consistency in geometry, coordinate reference systems, and attribute schema. Key preprocessing steps included harmonization of geometry types, removal of duplicate or overlapping features, and alignment of attribute structures across layers. Each feature was enriched with essential fields, including a unique identifier (feature_uid), a hierarchical reference (parent_uid), and a feature classification (feature_type). This standardization ensured structural compatibility and enabled consistent linkage across datasets.

4.4 Hierarchical UID Assignment

A hierarchical Unique Identifier (UID) system was applied to all spatial features to establish parent–child relationships and enable cross-layer integration. Each UID encodes feature type, hierarchical lineage, and a sequential identifier, allowing deterministic identification of features within their respective parent scope. The UID assignment process ensures uniqueness, preserves lineage, and provides a consistent reference framework for linking nodes and generating relationships. This structure forms the basis for graph construction and supports subsequent validation processes.

4.5 Graph Construction

The GeoGraphJSON graph is constructed by transforming spatial features into nodes and generating explicit edges based on defined relationship rules.

4.5.1 Node Generation: Each spatial feature is converted into a node containing its unique identifier, geometry, attributes, and hierarchical reference. This ensures that all entities are consistently represented within the graph structure.

4.5.2 Edge Generation: Edges are generated using rule-based logic, including containment edges for hierarchical inclusion, adjacency edges for spatial proximity, boundary edges for shared interfaces, functional edges for service linkages, and administrative edges for governance relationships.

Each edge is assigned a unique identifier, type, and source target linkage. Both directed and undirected relationships are supported, enabling representation of structural and functional connectivity across layers.

4.6 Validation Framework

The validation framework includes UID validation for identifier consistency, parent–child validation for hierarchical correctness, network connectivity validation for structural integrity, and edge consistency checks to ensure valid relationships and eliminate redundancy. These validation steps ensure that the resulting graph maintains logical consistency and supports reliable analytical operations.

4.7 Integration and Processing Pipeline

The complete workflow consists of five main stages: data ingestion, graph construction, validation and quality assurance, graph analytics, and export. Spatial datasets are first imported and standardized, followed by node and edge generation. Validation checks are then applied to ensure data integrity before analytical metrics are computed.

Finally, the dataset is exported in GeoGraphJSON and other compatible formats for visualization and analysis. The workflow is designed to integrate seamlessly with GIS environments and graph analysis tools, supporting both spatial processing and network-based analytics.

4.8 Summary

The proposed methodology provides a systematic approach for transforming conventional GIS datasets into a graph-enhanced spatial model. By integrating hierarchical UID structuring, rule-based edge generation, and multi-stage validation, the workflow ensures data consistency, relational integrity, and analytical readiness. This implementation framework establishes the foundation for evaluating GeoGraphJSON in real-world scenarios.

5. Evaluation Framework

5.1 Overview

A structured evaluation framework was developed to assess the integrity, consistency, and analytical capability of the GeoGraphJSON model. Unlike traditional assessments focused solely on geometric accuracy, this evaluation considers structural validity, topological consistency, and graph-based analytical behavior. The framework validates hierarchical relationships, network connectivity, and the ability of the model to represent realistic urban structures.

5.2 Structural and Topological Validation

Structural validation confirmed that all spatial entities adhered to the defined schema and hierarchical rules. The hierarchical UID system ensured consistent identification and traceable lineage across all layers, with no duplicate or malformed identifiers detected. Parent–child relationships were verified against spatial containment, ensuring that all features were correctly linked within the hierarchy.

Topological validation further confirmed the correctness of spatial connectivity. Road segments and nodes were fully connected within the network, forming a coherent structure suitable for traversal and analysis. Edge validation ensured that all relationships referenced valid nodes, with consistent edge types and no redundant or missing connections. These checks demonstrate that the GeoGraphJSON model maintains both structural and relational integrity.

5.3 Graph-Based Analytical Evaluation

Graph-theoretic metrics such as degree, density, clustering coefficient, and modularity are widely used to characterize the structure and connectivity of complex networks. Real-world networks often exhibit heterogeneous connectivity patterns influenced by growth and preferential attachment mechanisms (Barabási and Albert, 1999). In this study, the GeoGraphJSON dataset was evaluated using these metrics to assess its ability to represent realistic urban connectivity patterns. The resulting graph exhibited the following characteristics:

– Average degree ≈ 4.7
– Graph density ≈ 0.01
– Clustering coefficient ≈ 0.41
– Modularity ≈ 0.50

These values indicate a sparse yet cohesive network, consistent with real-world urban systems where localized clusters are interconnected through key nodes such as parcels and intersections. All nodes were contained within the largest connected component, confirming complete network connectivity. Community detection revealed distinct clusters corresponding to localized urban zones, demonstrating the model’s ability to capture spatial organization without predefined zoning. These results confirm that GeoGraphJSON preserves both hierarchical structure and topological realism, enabling meaningful spatial reasoning and clustering.

5.4 Functional and Scalability Assessment

The model was further evaluated for its ability to support spatial queries and analytical operations. GeoGraphJSON enables direct traversal of hierarchical relationships, identification of adjacency without runtime spatial joins, and extraction of functional linkages across layers. This reduces computational overhead compared to traditional GIS workflows where relationships must be derived dynamically. From a scalability perspective, the model supports incremental data expansion through its modular node–edge structure and hierarchical UID system. The observed sparse network characteristics suggest that the approach can scale to larger datasets while maintaining structural coherence.

5.5 Summary

The evaluation confirms that GeoGraphJSON maintains hierarchical and relational integrity, produces a topologically consistent graph, and enables efficient spatial and relational querying. These results demonstrate the feasibility of a unified spatial–graph framework, supporting scalable urban analytics and GeoAI applications.

6. Comparative Analysis

6.1 Overview

This section compares GeoGraphJSON with existing geospatial and graph-based data models in terms of geometry representation, relationship modeling, semantic capability, and analytical performance, highlighting its role in bridging GIS and graph-based approaches.

6.2 Geometry-Centric vs Graph-Based Models

Traditional geospatial formats such as GeoJSON, Shapefile, and GeoPackage support efficient storage and visualization of spatial geometry but lack explicit relationship representation. Relationships such as containment and adjacency must be derived dynamically, leading to repeated computation and increased overhead. CityGML and CityJSON introduce hierarchical and semantic structures but remain complex for lightweight and large-scale applications.

Graph-based models (e.g., Neo4j, NetworkX) explicitly represent relationships using nodes and edges, enabling efficient connectivity analysis and reasoning. However, they typically lack native support for spatial geometry, requiring transformation from GIS datasets and reducing interoperability. GeoGraphJSON integrates these paradigms by embedding graph connectivity within a GeoJSON-compatible structure, enabling both spatial and relational queries within a single dataset.

6.3 Comparative Summary

As shown in Table 2, GeoGraphJSON combines the strengths of geometry-centric and graph-based approaches by retaining geometric compatibility while enabling explicit relationship modeling, semantic representation, and efficient analytical processing.

Criteria	Geo JSON	City GML	Graph Models	GeoGraph JSON
Geometry	✓	✓	✕	✓
Relationships	✕	◐	✓	✓
Semantic	○	●	◐	●
Hierarchy	○	✓	✓	✓
Complexity	L	H	M	L–M
Interoperability	H	M	L	H
Query	L	M	H	H
AI Readiness	○	◐	●	●
Storage	H	M	M	M

Table 2: Spatial–Graph Model Comparison

Legend: ● Strong, ◐ Moderate, ○ Limited, ✓ Supported, ✕ Not Supported, L = Low, M = Medium, H = High

6.4 Advantages and Trade-offs

GeoGraphJSON integrates geometry, relationships, and semantics within a unified framework, enabling explicit relationship encoding, improved analytical efficiency, and compatibility with GIS and graph systems. However, explicit edge storage increases data volume, and updates to spatial features require corresponding updates to relationships. As a proposed model, further large-scale validation and adoption are required.

6.5 Summary

GeoGraphJSON bridges geometry-centric and graph-based models, enabling efficient querying, enhanced analytical capability, and interoperability across systems. This unified approach supports advanced spatial analytics and AI-driven applications.

7. Results and Analytical Insights

7.1 Dataset Overview and Graph Structure

The GeoGraphJSON framework was evaluated using a large-scale urban dataset representing a district-level integration of spatial layers. The dataset includes administrative hierarchies, transportation networks, and urban assets, structured through hierarchical Unique Identifiers and explicit edge relationships.

The dataset comprises administrative layers (districts, blocks, parcels, and buildings), transportation layers (road centerlines, segments, and nodes; bus routes, bus stops, metro lines and metro stations), and urban assets (service points, street lights, and trees). The resulting graph contains approximately 10,859 nodes and 13,733 edges representing containment, adjacency, functional, and administrative relationships.

This multi-layer structure enables a unified spatial–graph representation in which hierarchical organization and network connectivity are explicitly encoded. As presented in Table 3, the dataset exhibits a sparse structure with a single dominant connected component, along with moderate clustering and modularity, reflecting realistic urban connectivity patterns.

Category	Value
Study Area	Riyadh Urban Dataset
Total Nodes	10,859
Total Edges	13,733
Graph Type	Directed (mixed edge types)
Connected Components	1 (largest component)
Average Degree	~2.5 (undirected equivalent)
Graph Density	Low (sparse network)
Clustering Coefficient	Moderate (localized clustering)
Modularity	Moderate
Edge Types	Containment, Adjacency, Functional, Administrative
Node Types	Parcels, Buildings, Road Nodes

Table 3. Summary of GeoGraphJSON Dataset and Graph Characteristics

https://doi.org/10.5194/isprs-archives-XLIX-B4-2026-3-2026 | © Author(s) 2026. CC BY 4.0 License.

7.2 Network Characteristics and Connectivity

The generated graph exhibits structural properties consistent with real-world urban spatial networks. The network is sparse yet highly connected, reflecting the constrained nature of spatial systems where connectivity is influenced by geographic proximity and functional relationships.

The graph forms a dominant connected component, indicating strong integration across administrative and physical layers. Nodes maintain limited local connectivity while still contributing to global network cohesion. Furthermore, the combination of directed and undirected edges enables simultaneous representation of hierarchical dependencies and functional flows. These characteristics demonstrate that GeoGraphJSON successfully captures both structural and operational aspects of urban systems within a unified graph framework.

7.3 Node Degree Distribution

The node-degree distribution follows a right-skewed pattern, which is a common characteristic of spatial networks. As illustrated in Figure 5, a majority of nodes, such as buildings, service points, and trees, exhibit low degree values, reflecting localized interactions, whereas a smaller subset of nodes, including parcels and road intersections, show higher degree values and act as connectivity hubs.

This distribution reflects the hierarchical and spatial organization of urban environments, where low-degree nodes correspond to terminal or dependent entities, and high-degree nodes represent structurally important elements linking multiple spatial components. The presence of such heterogeneity indicates that the GeoGraphJSON model preserves realistic spatial topology and connectivity patterns.

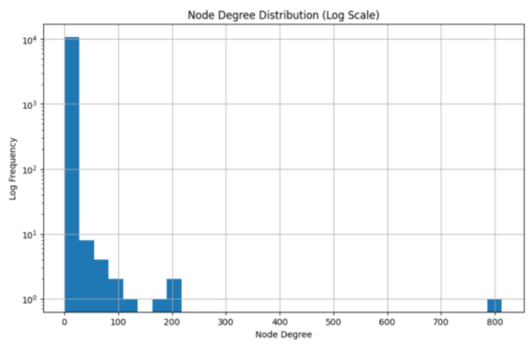

Figure 5. Node Degree Distribution of GeoGraphJSON Graph

7.4 Centrality and Node Influence

Graph centrality measures were applied to identify structurally significant nodes within the network. Nodes with high degree centrality correspond to parcels or intersections with multiple adjacencies, while nodes with high betweenness centrality act as bridges between clusters, facilitating spatial connectivity.

These nodes represent critical elements within the urban structure, including parcels with multiple road frontages, intersections connecting different functional zones, and centrally located parcels within dense blocks. The combined use of centrality metrics highlights spatial corridors of influence, which are essential for applications such as accessibility analysis, infrastructure planning, and service optimization.

7.5 Community Detection and Spatial Clustering

Community detection using the Louvain algorithm reveals distinct clusters within the network, corresponding to localized spatial groupings and functional zones. These communities represent parcel–building–road groupings within neighborhoods, localized infrastructure clusters, and spatially cohesive subgraphs reflecting urban organization.

The emergence of these clusters without predefined zoning attributes demonstrates the capability of GeoGraphJSON to support unsupervised spatial analysis, enabling discovery of functional patterns directly from data. This behavior highlights the capability of GeoGraphJSON to capture intrinsic spatial structure, enabling data-driven identification of urban functional zones through graph-based clustering.

7.6 Spatial–Graph Correlation Analysis

A correlation analysis between spatial attributes and graph-based metrics was conducted to evaluate the relationship between geometry and connectivity. The results indicate a positive correlation between parcel size and connectivity, suggesting that larger spatial units tend to have higher degrees and centrality. Moderate correlation among centrality measures reflects localized network efficiency, while a strong alignment between spatial hierarchy and graph structure confirms consistency between geometric and relational representations. These findings demonstrate that spatial characteristics and graph connectivity are intrinsically linked, reinforcing the validity of integrating geometry and topology within a unified data model.

7.7 Analytical Interpretation

The results indicate that the GeoGraphJSON model effectively captures hierarchical structure, spatial connectivity, and functional relationships within urban systems. The observed patterns demonstrate the ability of the model to represent complex spatial interactions in a unified graph framework.

7.8 Scope and Limitations

The presented results demonstrate the applicability of GeoGraphJSON to a large-scale urban dataset; however, certain limitations remain. The evaluation is conducted within a single metropolitan context (Riyadh), some attributes are derived or standardized for integration purposes, and performance benchmarking at multi-city or national scale remains to be explored. Despite these limitations, the current implementation demonstrates the model’s ability to scale beyond small area and operate effectively on complex, multi-layer urban datasets.

7.9 Summary

The results demonstrate that GeoGraphJSON supports realistic urban network representation and analysis.

8. Discussion

The results demonstrate that GeoGraphJSON provides a unified representation of spatial entities and their relationships within a single data structure. Unlike conventional GIS models, where relationships must be derived dynamically, GeoGraphJSON encodes these connections explicitly, enabling direct traversal and analytical operations. This approach transforms spatial datasets into interconnected structures that support both spatial analysis and relational reasoning.

The graph-based representation enhances analytical capability by enabling direct application of network analysis techniques. Metrics such as degree, clustering coefficient, and modularity reveal structural characteristics of urban systems that are not readily accessible in traditional workflows. The observed connectivity patterns and community structures confirm that the model captures realistic urban behavior, where certain entities function as hubs while others remain locally constrained. The presence of these connectivity hubs reflects heterogeneous connectivity patterns commonly observed in complex networks, often associated with growth and preferential attachment processes (Barabási and Albert, 1999). These findings are consistent with the principles of spatial networks, where geographic constraints and the cost of connections influence network structure, connectivity, and efficiency (Barthelemy, 2011). The framework is designed to scale to larger datasets through its modular node–edge structure and hierarchical UID system, allowing incremental data expansion without restructuring. Compatibility with graph databases further supports efficient processing of large-scale networks. However, validation at city and regional scales requires further investigation.

A key trade-off lies in the explicit storage of relationships. While this reduces computational overhead and improves analytical efficiency, it increases data volume and introduces challenges in maintaining consistency during updates. Strategies such as selective edge generation and hybrid computation can mitigate these limitations.

GeoGraphJSON maintains strong interoperability by remaining compatible with standard GIS formats while enabling seamless integration with graph analytics and AI workflows. This dual capability positions it as a bridge between traditional geospatial systems and emerging GeoAI applications, including digital twins and graph-based learning models. Although the current evaluation is constrained by dataset scope and attribute completeness, the framework demonstrates strong potential for real-world deployment. Overall, GeoGraphJSON advances geospatial data modeling by unifying geometry and connectivity, enabling scalable analysis and supporting the transition toward intelligent, AI-driven spatial systems.

9. Conclusions

This paper presented GeoGraphJSON, a lightweight semantic data model that integrates spatial geometry with graph-based connectivity to support advanced spatial reasoning and GeoAI applications. By extending geometry-centric formats with explicit, typed relationships, the model enables unified representation of spatial entities and their interactions within a single interoperable structure.

A hierarchical UID system ensures consistent lineage, traceability, and cross-layer linkage, while the edge framework captures containment, adjacency, functional, administrative, and semantic relationships in a machine-interpretable form. The proposed methodology demonstrates how multi-layer geospatial datasets can be transformed into graph-enhanced representations through standardization, UID assignment, rule-based edge generation, and validation.

The implementation framework produced a coherent and topologically consistent graph, with network metrics confirming realistic urban patterns and identifiable connectivity structures. Comparative analysis shows that GeoGraphJSON effectively bridges the gap between geometry-centric GIS formats and graph-based models, enabling efficient querying and direct integration with graph analytics workflows.

While explicit edge storage increases data volume, it significantly improves analytical efficiency, highlighting a practical trade-off between storage and computation. Although the current implementation is limited to a study dataset, the results validate the feasibility and effectiveness of the proposed approach. Future work will focus on large-scale deployment, real-time data integration, and extension toward semantic knowledge graphs and predictive GeoAI models. Overall, GeoGraphJSON provides a scalable and extensible foundation for transforming geospatial data into interconnected, analysis-ready structures for next-generation urban analytics.

References

Barabási, A.-L., Albert, R., 1999. Emergence of scaling in random networks. Science, 286(5439), 509–512.

Barthelemy, M., 2011. Spatial networks. Physics Reports, 499(1–3), 1–101.

Butler, H., Daly, M., Doyle, A., Gillies, S., Schaub, T., Schmidt, C., 2016. The GeoJSON Format. RFC 7946, IETF.

Chen, H., 2018. A graph database model for knowledge extracted from place descriptions. ISPRS Int. J. Geo-Information, 7(6), 221.

ISO, 2014. ISO 19115-1:2014 Geographic information — Metadata — Part 1: Fundamentals. International Organization for Standardization.

Ledoux, H., Arroyo Ohori, K., Kumar, K., Dukai, B., Labetski, A., Vitalis, S., 2019. CityJSON: A compact and easy-to-use encoding of the CityGML data model. Open Geospatial Data, Software and Standards, 4(1), 4.

OGC, 2012. GeoSPARQL – A Geographic Query Language for RDF Data. Open Geospatial Consortium.

OGC, 2021. OGC API – Features – Part 1: Core. Open Geospatial Consortium.