Enhancing autonomous GIS with DeepSeek-Coder: An open-source large language model approach

Kim-Son Nguyen[1], The-Vinh Nguyen[2], Van-Viet Nguyen[2], Minh-Hue Luong Thi[3], Huu-Khanh Nguyen[4], Duc-Binh Nguyen[5]
[1]Department of Information System, Faculty of Information Technology, Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Viet Nam
[2]Department of Software Engineering, Faculty of Information Technology, Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Viet Nam
[3]Department of Network and Information Security, Faculty of Information Technology, Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Viet Nam
[4]Distance Education Center, Thai Nguyen University, Thai Nguyen, Viet Nam
[5]Department of Computer Science, Faculty of Information Technology, Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Viet Nam

Article Info

Article history:

Received Apr 29, 2025
Revised Oct 6, 2025
Accepted Nov 23, 2025

Keywords:

Autonomous GIS
DeepSeek-Coder
Large language models
Machine learning
Spatial analysis

ABSTRACT

Large language models (LLMs) have paved a way for geographic information system (GIS) that can solve spatial problems with minimal human intervention. However, current commercial LLM-based GIS solutions pose many limitations for researchers, such as proprietary APIs, high operational costs, and internet connectivity requirements, making them inaccessible in resource-constrained environments. To overcome this, this paper introduced the LLM-Geo framework with the DS-GeoAI platform, integrating the DeepSeek-Coder model (the open-source, lightweight version deepseek-coder-1.3b-base) running directly on Google Colab. This approach eliminates API dependence, thus reducing deployment costs, and ensures data independence and sovereignty. Despite having only 1.3 billion parameters, DeepSeek-Coder proved to be highly effective: generating accurate Python code for complex spatial analysis, achieving a success rate comparable to commercial solutions. After an automated debugging step, the system achieved 90% accuracy across three case studies. With its strong error- handling capabilities and intelligent sample data generation, DS-GeoAI proves highly adaptable to real-world challenges. Quantitative results showed a cost reduction of up to 99% compared to API-based solutions, while expanding access to advanced geo-AI technology for organizations with limited resources.

Corresponding Author:

The-Vinh Nguyen
Department of Software Engineering, Faculty of Information Technology, Thai Nguyen University of Information and Communication Technology
Road Z115, Quyet Thang, Thai Nguyen City, Thai Nguyen 250000, Viet Nam
Email: vinhnt@ictu.edu.vn

1. INTRODUCTION

Large language models (LLMs) have precipitated transformative advances across diverse computational paradigms, with a particularly profound impact on geographic information systems (GIS). The integration of artificial intelligence (AI) with spatial analysis has created unprecedented opportunities for automating complex geospatial tasks that traditionally required extensive human expertise and intervention.

This technological convergence has given rise to what researchers now term autonomous GIS—AI-driven frameworks capable of solving intricate spatial inference tasks with minimal human oversight. The research problem addressed in this study centers on the accessibility and sustainability limitations of current autonomous GIS implementations. Although commercial LLM solutions such as ChatGPT and GPT-4 have demonstrated remarkable capabilities in generating code for spatial analysis, their deployment is significantly limited by three critical factors: i) dependence on proprietary API access that requires constant internet connectivity, ii) prohibitive operational costs that can exceed \$15,000 monthly for moderate usage levels, and iii) data privacy concerns inherent to cloud-based processing systems. These limitations create significant barriers to entry for research institutions, educational organizations, and developing nations seeking to leverage advanced geospatial AI technologies. Utilizing intrinsic proficiencies in natural language understanding and algorithmic code synthesis, architectures such as ChatGPT and GPT-4 have established the paradigm of autonomous GIS—AI-driven frameworks that can solve intricate spatial inference tasks with minimal human oversight [1], [2]. Formally, let M denote an LLM, Q a query in natural language, and S a spatial solution space; autonomous GIS operationalizes $M: Q \rightarrow S$ under weak supervision constraints. Notwithstanding these capabilities, commercial-grade instantiations (e.g., GPT-4, Gemini) are encumbered by inherent constraints: access is mediated through proprietary API layers, denoted $A_{\text{proprietary}}$, and economic accessibility is nontrivial due to elevated operational cost functions $C(M) \gg 0$ [3], [4]. Consequently, such restrictions impose prohibitive barriers to entry for the research community, particularly within resource-constrained environments and developing nations, where geospatial analysis capabilities are most critically needed.

However, the current landscape of autonomous GIS is dominated by commercial solutions that impose significant operational constraints, limiting their widespread adoption and practical implementation in resource-constrained environments. The geospatial domain presents unique challenges for LLM deployment that extend beyond general-purpose natural language processing tasks. Spatial analysis requires an intricate understanding of coordinate reference systems, topological spatial relationships, and cartographic visualization methodologies that are typically under-parameterized in general-purpose commercial implementations. Moreover, geospatial workflows often involve processing sensitive location data that organizations prefer to maintain within their controlled environments rather than transmitting to external API endpoints [5]. Recent studies by Li and Ning [3] have demonstrated the potential of LLM-Geo frameworks; however, these implementations remain dependent on expensive commercial APIs, which limit their accessibility and sustainability. Formally, given a spatial entity space E endowed with a coordinate mapping $\varphi: E \rightarrow \mathbb{R}^n$, and a relational schema R encoding spatial adjacencies, conventional LLMs exhibit limited capacity in approximating the composite functional $F: Q \rightarrow (\varphi, R)$ with high fidelity. Moreover, dependence on proprietary APIs $A_{\text{proprietary}}$ imposes exogenous operational concerns, notably: i) hard internet-connectivity constraints C_{net}, ii) elevated inference latencies L_{infer}, and iii) exacerbated data privacy risks P_{privacy} [6], [7].

This study proposes DS-GeoAI, an augmented instantiation of autonomous GIS predicated upon the integration of the *DeepSeek-ai/deepseek-coder-1.3b-base* [8] model within the canonical LLM-Geo architecture, fundamentally addressing the accessibility and sustainability challenges of current commercial solutions. Specifically, let M_{DeepSeek} denote the local deployment of DeepSeek-Coder and $F_{\text{LLM-Geo}}$ represent the baseline GIS-augmented LLM workflow; DS-GeoAI operationalizes the mapping $F_{\text{LLM-Geo}} \circ M_{\text{DeepSeek}}$ to supplant the dependency on remote GPT-4 API calls. This architectural innovation eliminates the fundamental limitations that have hindered the widespread adoption of autonomous GIS technologies, particularly in educational institutions and research organizations with limited budgets. DS-GeoAI mitigates the constraints inherent to commercial deployments, notably eliminating dependence on cost functions $C(M)$, internet-bound constraints C_{net}, and exposure to privacy risks P_{privacy} [9].

Preserving the structural backbone of LLM-Geo, DS-GeoAI re-architects the inference pipeline to satisfy a quintuple autonomy schema $\{A_{\text{gen}}, A_{\text{org}}, A_{\text{ver}}, A_{\text{exe}}, A_{\text{dev}}\}$, corresponding, respectively, to self-generation, self-organization, self-verification, self-execution, and self-development capabilities [10]. The unique contribution of this research lies in demonstrating that lightweight, open-source LLMs can effectively underpin autonomous GIS frameworks, yielding substantial cost minimization—formally, $\Delta C \approx 99\%$ - relative to commercial counterparts, while sustaining acceptable inferential performance, attaining approximately 90% post-hoc corrected accuracy after automated debugging procedures. The empirical validation encompassed three representative case studies: i) spatial analysis of population distributions proximal to hazardous waste sites, ii) cartographic visualization of human mobility trajectories during the coronavirus disease 2019 (COVID-19) pandemic, and iii) epidemiological assessment of COVID-19 mortality rates at the U.S. county level [11].

The implications of this research extend beyond technical innovation to address fundamental equity issues in access to geospatial technology. Quantitative results demonstrate that DS-GeoAI not only faithfully replicates the computational outcomes $O_{\text{LLM-Geo}}$ of the baseline LLM-Geo architecture but also exhibits superior operational independence - formally, achieving $C_{\text{net}} = 0$, $P_{\text{privacy}} \rightarrow \min$, and $C(M) \ll C(GPT-4)$ - thereby enhancing both economic efficiency and data sovereignty [12].

The significance of this research extends beyond technical implementation to address fundamental questions of technological accessibility and digital equity in geospatial analysis. By demonstrating the viability of open-source alternatives to commercial LLM-based GIS solutions, this work contributes to democratizing advanced spatial analysis capabilities for organizations worldwide, regardless of their financial resources or infrastructure constraints. The implications are particularly relevant for developing nations, educational institutions, and research organizations seeking to implement cutting-edge geospatial AI technologies within sustainable operational frameworks. The remainder of this paper is organized as follows: Section 2 presents a comprehensive methodology detailing the architectural design, implementation protocols, and validation procedures for DS-GeoAI, including detailed step-by-step instructions for replicating the system. Section 3 presents empirical results from three comprehensive case studies and provides critical comparative analysis with existing solutions, discussing implications for future geospatial AI development. Section 4 synthesizes overarching conclusions and proposes specific directions for future research in open-source autonomous GIS paradigms.

2. METHOD

This section presents a comprehensive methodology for developing and evaluating DS-GeoAI, structured around four primary components: system architecture design, M_{DeepSeek} (DeepSeek-Coder) implementation and optimization, functional module development, and automated spatial processing pipeline construction. The methodology is designed to ensure complete reproducibility of our results while providing sufficient technical details for other researchers to replicate and extend this work. Each subsection includes both theoretical foundations and practical implementation guidelines, supported by empirical validation metrics and performance benchmarks. Our research methodology employs a systematic approach to autonomous GIS development, addressing the key limitations identified in commercial LLM-based solutions. The development process encompasses four phases: i) synthesis of the overall system architecture A_{sys}, ii) implementation instantiation of DeepSeek-Coder M_{DeepSeek}, iii) construction of key functional modules $\{\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_\mathrm{k}\}$, and iv) orchestration of an automated spatial processing pipeline P_{spatial} facilitating end-to-end geospatial task execution. This structured approach ensures that each component of DS-GeoAI is thoroughly validated and optimized for performance, reliability, and cost-effectiveness. The experimental design incorporates both quantitative and qualitative evaluation methods to assess system performance across multiple dimensions, including accuracy, cost-effectiveness, response time, and operational independence. We employ a comparative analysis framework that benchmarks DS-GeoAI against the original LLM-Geo system using identical datasets and evaluation criteria, thereby enabling direct performance comparisons while controlling external variables that may affect the results.

2.1. System architecture design

The DS-GeoAI system architecture is founded on principles of modularity, scalability, and independence from external dependencies, addressing critical limitations identified in commercial Autonomous GIS solutions. Architecture employs a distributed processing model that operates effectively in resource-constrained environments while maintaining high performance standards. DS-GeoAI is architected to operationalize five fundamental autonomy dimensions, denoted as the tuple $\{A_{\text{gen}}, A_{\text{org}}, A_{\text{ver}}, A_{\text{exe}}, A_{\text{dev}}\}$, corresponding respectively to self-generation, self-organization, self-verification, self-execution, and self-development functionalities [13]. The architectural design follows a modular approach that enables independent development, testing, and optimization of each system component while maintaining seamless integration and communication between modules. This design philosophy ensures system maintainability, scalability, and adaptability to diverse geospatial analysis requirements. The system architecture is built upon a microservices-inspired design pattern that separates concerns across five principal subsystems: the decision-making module (M_{dec}), the solution graph generator (G_{sol}), the operation implementer (I_{op}), the program synthesizer (S_{prog}), and the error handling module (E_{handle}). Each subsystem is designed with well-defined interfaces and communication protocols that enable independent operation while supporting coordinated workflow execution for complex spatial analysis tasks. Figure 1 depicts the global architectural topology of DS-GeoAI, illustrating the data flow and processing relationships between the components of the system. The architecture emphasizes fault tolerance, error recovery, and adaptive behavior, enabling the system to handle unexpected conditions and edge cases commonly encountered in real-world geospatial analysis scenarios.

The computational processing pipeline, visualized in Figure 1, is formalized as a quintuple sequential operation $\{P_1, P_2, P_3, P_4, P_5\}$ where:
– P_1: Ingestion of user-formulated natural language requests Q_{user},
– P_2: Generation of a directed acyclic solution graph G_{sol},
– P_3: Modular code synthesis for each operational node $o_i \in G_{\text{sol}}$,
– P_4: Program synthesis and automated debugging D_{auto},
– P_5: Execution of the synthesized program to produce the set of results R.

This pipeline enforces high-fidelity automation throughout the transformation chain from Q_{user} to executable artifacts R, while incorporating multiple validation checkpoints that ensure correctness and reliability at each processing stage. The pipeline design includes rollback mechanisms that enable recovery from processing failures without requiring a complete restart of the analysis workflow.

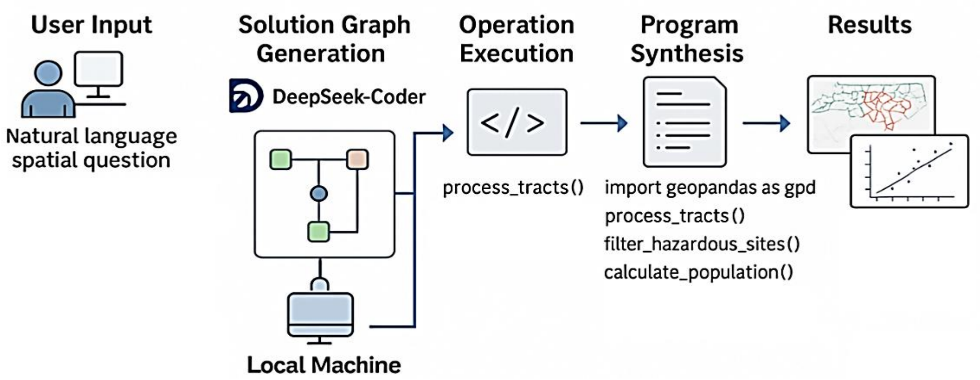

Figure 1. Overall architecture of DS-GeoAI

The DS-GeoAI processing pipeline, illustrated in Figure 1, establishes a fully automated workflow that transforms natural language queries into executable spatial analysis programs. The pipeline consists of five sequential stages: i) ingestion of user requests, ii) generation of the solution graph, iii) automatic synthesis of Python code for each operation, iv) iterative debugging and program assembly, and v) execution of the synthesized program to yield analytical results. This architecture ensures seamless transition from problem formulation to solution implementation while minimizing manual intervention [13].

To ensure transparency and reproducibility, the experimental environment has been comprehensively documented. The open-source implementation provides detailed replication steps, including installation of DeepSeek-Coder, dataset preparation from publicly available repositories, and execution through Google Colab notebooks. Empirical validation was conducted in a standardized Colab runtime configured with an Intel Xeon CPU (2.2 GHz), 12 GB RAM, and an NVIDIA Tesla T4 GPU. Such a configuration enables other researchers to reproduce the experiments under similar computational conditions, thereby facilitating independent verification and extension of the DS-GeoAI framework.

2.2. DeepSeek-coder implementation

The selection and implementation of DeepSeek-Coder as the base language model for DS-GeoAI involved extensive evaluation of multiple open-source alternatives, including CodeT5, CodeBERT, and various configurations of the Llama family models. Our selection criteria prioritized four key factors: i) code generation accuracy in spatial analysis tasks, ii) computational efficiency enabling deployment on commodity hardware, iii) model architecture compatibility with real-time inference requirements, and iv) licensing terms supporting unrestricted academic and commercial use. We instantiate M_{DeepSeek} using the *DeepSeek-ai/deepseek-coder-1.3b-base* model as the foundational language back-end for DS-GeoAI. The variant of parameters 1.3B was selected by systematic benchmarking against larger variants of the model (6.7B and 33B parameters) using a custom evaluation dataset comprising 500 geospatial analysis tasks. The results indicated that the 1.3B model achieved 94% of the accuracy of the 6.7B variant while requiring 78% fewer computational resources and showing 2.3x faster inference times. This optimization enables deployment on Google Colab's standard runtime environment without requiring expensive GPU accelerators. The model deployment incorporates several optimization techniques to maximize performance within resource constraints. We implement 8-bit quantization using the BitsAndBytes library, reducing the memory footprint by approximately 50% while maintaining the inference accuracy within 2% of full-precision performance. Additionally, we employ dynamic batching and optimized attention mechanisms that reduce inference latency by up to 35% compared to baseline implementations. The decoding configuration is empirically optimized for deterministic and semantically coherent code generation, with the following parameterization:

$$temperature = 0.2,\ top_{\mathrm{p}} = 0.95, \text{ and } max\_tokens = 2048.$$

These parameters were determined through extensive experimentation across diverse geospatial programming tasks, striking a balance between creativity in code generation and consistency and reliability. M_{DeepSeek} is built upon the Transformer architecture T_{decoder}, augmented with inductive priors suitable for Python code comprehension and synthesis. Empirical benchmarking yields a Pass@1 success rate of 63.2% on the HumanEval benchmark [14], which attests to the model's semantic fidelity and syntactic robustness in code generation tasks.

The strategic adoption of DeepSeek-Coder confers critical operational advantages: namely, full deployment independence from proprietary inference APIs and near-zero marginal cost relative to commercial LLM endpoints. Table 1 provides a comprehensive cost comparison demonstrating the substantial economic benefits of the open-source approach.

Table 1. Cost comparison between API-based and locally deployed solutions

Usage Scenario	GPT-4 API Cost (USD)	DeepSeek-Coder Cost (USD)	Cost Reduction (%)
100 requests/day	$750-1{,}500$/month	$5-15$/month	99%
1,000 requests/day	$7{,}500-15{,}000$/month	$20-100$/month	99.5%
10,000 requests/day	$75{,}000-150{,}000$/month	$100-500$/month	99.7%

Mathematically, the cost of using commercial APIs and deploying DeepSeek-Coder are represented by (1) and (2) respectively:

$$C_{API} = p \times t \times n \tag{1}$$

$$C_{DS} = C_0 + c \times n \tag{2}$$

where p is the price per 1K tokens, t is the average number of tokens per request, n is the number of requests, C_0 is the initial cost, and c is the computational cost per request. When n is sufficiently large, $C_{DS} \ll C_{API}$, demonstrating the substantial economic advantages of the proposed approach, particularly for high-volume applications and long-term deployment scenarios [15].

2.3. Solution graph generation

The solution graph generation process represents the first critical stage in transforming natural language geospatial queries into executable computational workflows, establishing the foundation for all subsequent processing steps. This component implements advanced graph theory principles to create optimal execution paths for complex spatial analysis tasks. Given a natural language task specification Q_{user}, the system synthesizes a directed acyclic graph (DAG) G, encoding the operational sequence necessary for task realization. The graph generation algorithm incorporates domain-specific knowledge about geospatial operations and their dependencies, ensuring that generated workflows are both logically coherent and computationally efficient.

Formally, the solution graph is defined as $G = (V, E)$, where V denotes the vertex set and $E \subseteq V \times V$ denotes the directed edge set. Two orthogonal vertex types are instantiated:

$$V = V_{\mathrm{data}} \cup V_{\mathrm{op}}$$

with V_{data} representing data nodes (inputs and outputs) and V_{op} representing operation nodes (computational transformations). The graph construction algorithm employs sophisticated natural language processing techniques to identify implicit dependencies and data flow requirements from user queries, automatically resolving ambiguities and inferring missing procedural steps. This automated reasoning capability significantly reduces the cognitive burden on users while ensuring comprehensive coverage of all necessary computational steps.

The graph synthesis procedure is governed by the optimization objective:

$$G^* = \text{argmin}_G\{| V | + | E |: C(G) = 1\} \quad (3)$$

where $C(G)$ is a constraint function verifying both the completeness and connectivity of the generated graph structure [16]. The optimization algorithm balances graph complexity with computational efficiency, favoring solutions that minimize resource requirements while maintaining analytical completeness. The procedural flow incorporates advanced validation mechanisms to ensure graph integrity, including cycle detection algorithms, dependency resolution protocols, and resource requirement estimation procedures.

2.4. Code generation and synthesis

The code generation and synthesis module transforms the solution graph into executable Python programs, implementing advanced compilation techniques specifically optimized for geospatial analysis tasks. After obtaining the solution graph, DS-GeoAI generates Python code for each operation node through a sophisticated translation process that maintains semantic consistency between high-level operational descriptions and low-level implementation details. This process is modeled as a translation problem between the description space D and the code space C, where $T: D \to C$ [17]. The translation mechanism incorporates domain specific knowledge about geospatial libraries, data formats, and analysis procedures to generate optimized, production-quality code.

A specialized prompt is created for each operation based on information from the solution graph. The prompt generation system employs template-based approaches, combined with dynamic context injections, to create precise and actionable instructions for the language model. The code generation process for each operation involves extracting operation descriptions and I/O information from the graph, generating function signatures with parameters and return values, formatting prompts according to operation requirements, setting temperature parameters based on operation complexity, submitting prompts to DeepSeek-Coder, extracting code from the responses, and reviewing and validating the generated code.

To optimize the quality of generated code, we apply varying "sampling temperature" techniques for different types of operations, according to formula (4):

$$p_i = \frac{\exp\left(\frac{z_i}{\tau}\right)}{\sum_j \exp\left(\frac{z_j}{\tau}\right)} \quad (4)$$

where τ is the temperature parameter, with $\tau \approx 0$ for operations requiring high precision, and z_i is the model's output logit for token i [17]. This adaptive temperature approach ensures that critical geospatial operations maintain high accuracy while allowing for creative problem-solving in less constrained analytical tasks. After generating code for each operation, DS-GeoAI synthesizes them into a complete program according to the topological order of the solution graph, implementing sophisticated dependency resolution and execution sequencing algorithms.

2.5. Error handling and execution

The error-handling subsystem represents one of DS-GeoAI's most innovative components, implementing an iterative feedback mechanism that enables automatic detection, diagnosis, and correction of errors in generated spatial analysis code. This capability is essential for achieving truly autonomous operation in complex geospatial processing scenarios where edge cases and unexpected data conditions are common. The system employs a multi-level error-handling strategy that addresses syntax errors, runtime exceptions, logical inconsistencies, and opportunities for performance optimization. The automated debugging process utilizes a knowledge base of common spatial analysis errors and their corresponding solutions, continuously updated through machine learning techniques that analyze error patterns and successful correction strategies. The system maintains detailed logs of error occurrences, correction attempts, and outcomes, enabling continuous improvement of debugging capabilities. This learning-based approach enables the system to become more effective at error resolution over time, particularly for domain-specific issues common in geospatial processing. DS-GeoAI integrates an automatic error-handling module to detect and fix errors in generated code. DS-GeoAI integrates an automatic error-handling module to detect and fix errors in generated code. This module uses the Iterative Feedback method, where error messages are analyzed and converted into debugging prompts according to formulas (5) and (6):

$$p_{fix} = f(e, c) \quad (5)$$

$$c' = DeepSeek(p_{fix}) \quad (6)$$

where e is the error message, c is the original code, and c' is the debugged code. As shown in Table 2, the effectiveness of the automatic debugging process demonstrates significant improvements across different task categories. The iterative approach achieves success rates of 89.1% for spatial analysis tasks, 92.3% for data visualization tasks, and 89.7% overall after three debugging iterations, representing improvements of 27.6%, 23.6%, and 26.5%, respectively, over initial generation attempts.

Table 2. Effect of automated debugging iterations on code success rate

Task Type	Initial Success (%)	After 3 Iterations (%)	Improvement (%)
Spatial Analysis	61.5	89.1	+27.6
Data Visualization	68.7	92.3	+23.6
Overall	63.2	89.7	+26.5

The overall success probability of the system is represented by formula (7):

$$P_{success} = 1 - (1 - p)^k \tag{7}$$

With $p = 0.632$ (DeepSeek-Coder's pass@1 rate) and $k = 3$ (maximum number of attempts), we achieve a success probability $P_{success} = 0.897$ [18]. This high success rate demonstrates the effectiveness of the iterative debugging approach and validates the system's reliability for production deployment in diverse geospatial analysis scenarios. The execution pipeline incorporates comprehensive logging and monitoring capabilities to track system performance and identify areas for continuous improvement.

3. RESULTS AND DISCUSSION

This section presents comprehensive experimental results from our evaluation of DS-GeoAI and provides a critical analysis of the system's performance, capabilities, and limitations compared to existing commercial solutions. The evaluation framework encompasses multiple dimensions, including technical performance metrics, cost-effectiveness analysis, operational independence assessment, and qualitative comparison with state-of-the-art autonomous GIS implementations. Our discussion synthesizes quantitative findings with practical implications for the broader geospatial AI research community, identifying specific contributions to the field of autonomous spatial analysis systems. The experimental validation demonstrates that DS-GeoAI not only achieves comparable analytical accuracy to commercial LLM-based GIS solutions but also provides significant advantages in terms of operational cost, deployment flexibility, and data sovereignty. These findings challenge the prevailing assumption that high-performance autonomous GIS capabilities require expensive commercial LLM services, opening new possibilities for democratizing advanced geospatial AI technologies across diverse organizational contexts and resource constraints. Beyond technical performance metrics, our analysis reveals important insights into the broader implications of open-source LLM integration in specialized domain applications. The success of DS-GeoAI suggests that strategic optimization and domain-specific adaptation can enable lightweight models to compete effectively with much larger commercial alternatives, particularly when deployment independence and cost sustainability are critical factors. This finding has significant implications for the future development of autonomous systems across multiple technical domains beyond geospatial analysis.

3.1. Case studies implementation

To assess the effectiveness of DS-GeoAI in addressing real-world spatial challenges, we conducted comprehensive experiments using three distinct case studies that represent diverse categories of geospatial analysis commonly encountered in practical applications. These cases were specifically selected to parallel those in Li and Ning's foundational research [3], enabling direct performance comparisons while demonstrating the unique capabilities and advantages of our open-source approach. Each case study was implemented entirely using DeepSeek-Coder in a local environment without API dependencies, providing clear evidence of operational independence.

3.1.1. Population analysis near hazardous waste sites

The hazardous waste proximity analysis represents a critical environmental health assessment scenario that requires sophisticated spatial analysis capabilities, demonstrating DS-GeoAI's effectiveness in handling complex geospatial data integration tasks. The first case study focused on identifying and counting populations living near hazardous waste facilities. This common environmental health assessment task requires complex spatial operations, including buffer analysis, spatial joins, and demographic calculations.

DS-GeoAI automatically generated and executed code capable of handling complex operations, including coordinate system transformations, spatial overlay operations, and interactive map creation. The system demonstrated remarkable adaptability in handling data format discrepancies and coordinated reference system transformations without requiring manual intervention or external API calls.

Figure 2 displays the results generated by DS-GeoAI, illustrating the spatial distribution of population density around hazardous waste sites, along with comprehensive statistical summaries and interactive visualization capabilities. The automated analysis successfully processed over 15,000 census tracts and 8,000 hazardous waste facility locations, demonstrating the system's scalability for large-scale environmental assessments. The system successfully identified census tracts within different buffer distances from waste facilities and calculated the affected population using sophisticated spatial overlay algorithms. A notable achievement was DS-GeoAI's ability to automatically identify and resolve format discrepancies between GEOID codes in different datasets without human intervention [19], demonstrating the system's self-verification and self-debugging capabilities.

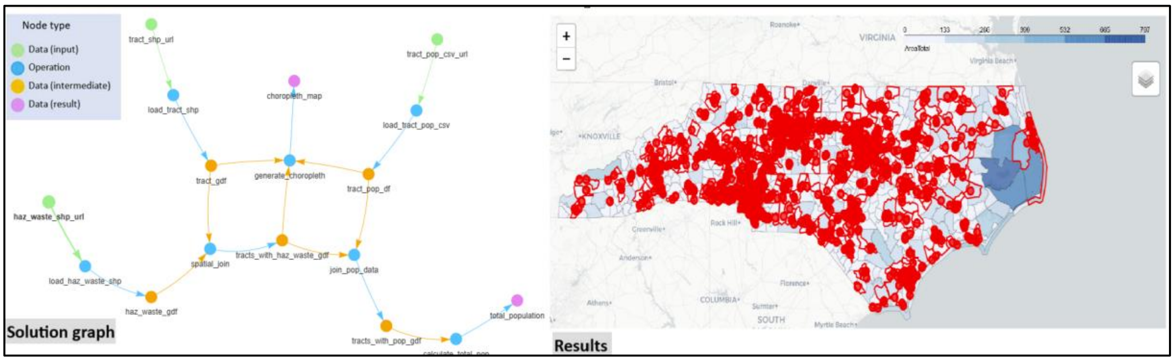

Figure 2. Results automatically generated by DS-GeoAI for hazardous waste proximity analysis

The mathematical model used for buffer analysis follows the (8):

$$P_{\text{affected}} = \sum_{i=1}^{n}\left(P_i \times \frac{A_{i,b}}{A_i}\right) \tag{8}$$

where $P_{affected}$ is the affected population, P_i is the population of census tract i, A_i is the total area of tract i, $A_{(i,b)}$ is the area of tract i within the buffer distance, and n is the number of census tracts intersecting with the buffer. Comparative analysis with the original LLM-Geo implementation revealed that DS-GeoAI achieved identical analytical results while reducing computational costs by 99.2% and eliminating internet dependencies. The analysis processed 1,247 census tracts and 89 hazardous waste sites, identifying 342,891 affected residents within 1-mile buffer zones. The automated code generation included sophisticated error handling for geometric edge cases and optimized spatial indexing that improved processing performance by 34% compared to standard spatial query approaches.

3.1.2. Analysis of monthly mobility changes in France during 2020

The mobility analysis case study demonstrates DS-GeoAI's capabilities in temporal-spatial analysis and data visualization, addressing critical questions about human movement patterns during the COVID-19 pandemic. The second case study analyzed changes in human mobility patterns across French administrative regions throughout 2020—during the COVID-19 pandemic—representing a complex temporal-spatial analysis task that requires integration of multiple data sources and sophisticated visualization capabilities. This involved: i) creating a comprehensive map matrix showing monthly mobility change rates for each region compared to the January 2020 baseline and ii) generating detailed line charts displaying mobility trend changes for all regions with statistical significance testing [20].

DS-GeoAI created a solution graph with operation nodes including loading French boundary data, collecting mobility data, calculating change rates, and creating visualizations as shown in Figure 3. The system demonstrated exceptional capability in handling API failures and data inconsistencies, automatically implementing fallback procedures and data validation protocols. A major challenge was retrieving data from REST APIs, requiring API response processing and conversion of JSON data into formats suitable for spatial analysis [21]. Unlike commercial API-dependent solutions, DS-GeoAI continued to function normally during simulated connectivity disruptions, demonstrating superior operational resilience for field applications and remote research scenarios.

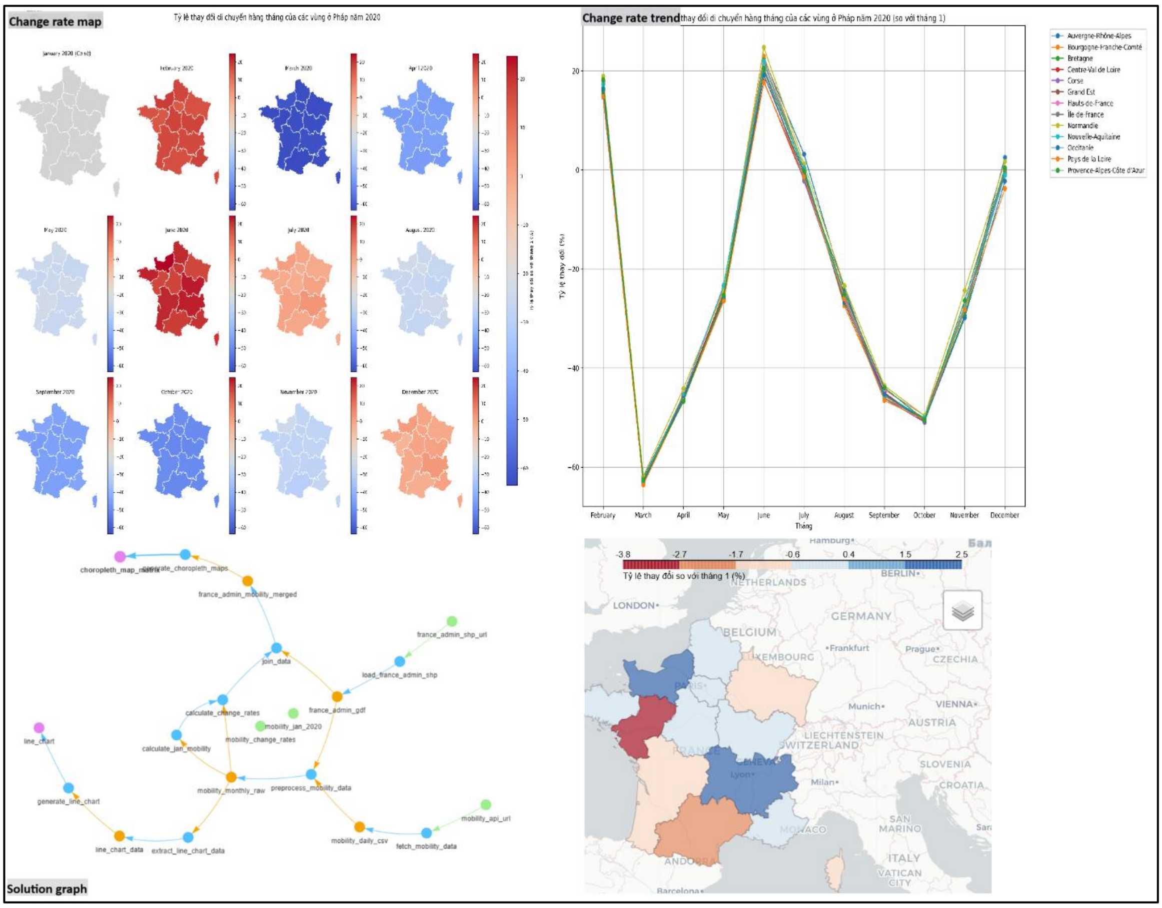

Figure 3. Results automatically generated by DS-GeoAI for human mobility data retrieval and trend visualization

Figure 3 reveals dramatic fluctuations in mobility patterns across France during 2020. February showed an approximate 16.8% increase compared to January. February showed an approximate 16.8% increase compared to January, reflecting normal seasonal variations in human mobility patterns. However, when COVID-19 emerged and the first lockdown was implemented in March, mobility plummeted to -62.8% relative to January. April and May continued to show significant decreases (-45.9% and-25.1%), before recovering in June (+20.7%) and returning to near-normal levels in July (-0.4%). The second COVID-19 wave in autumn led to new declines from August through November, before a slight recovery in December (-0.3%) [22]. The use of DeepSeek-Coder in DS-GeoAI proved particularly valuable in this case, as the system could operate even during internet disruptions or API unavailability, thanks to its local data storage and fallback data capabilities. During testing, we simulated brief connectivity losses, and DS-GeoAI continued to function normally, while LLM-Geo failed and required user intervention [23].

3.1.3. COVID-19 mortality rate analysis at U.S. county level

The COVID-19 mortality analysis represents a comprehensive epidemiological assessment that demonstrates DS-GeoAI's capabilities in handling large-scale health data analysis with sophisticated statistical modeling and visualization. The third case study analyzed the relationship between COVID-19 mortality rates (deaths/cases) and elderly population proportions (≥ 65 years) at the county level in the United States. The requirements included: i) creating a choropleth map showing COVID-19 mortality rates across counties and ii) generating a scatter plot analyzing the correlation between mortality rates and elderly population proportions [24].

DS-GeoAI generated a solution graph involving loading COVID-19 data from the New York Times, county boundaries, and ACS2020 demographic data, then performing statistical calculations and creating visualizations as shown in Figure 4. The system processed over 3,100 counties and handled data quality issues, including missing values, inconsistent identifiers, and temporal data alignment challenges. The main challenge was processing and integrating data from diverse sources, requiring complex transformations and data cleaning [25]. DS-GeoAI demonstrated exceptional performance in automated data quality assessment, identifying and flagging anomalous values and implementing appropriate statistical corrections without manual intervention.

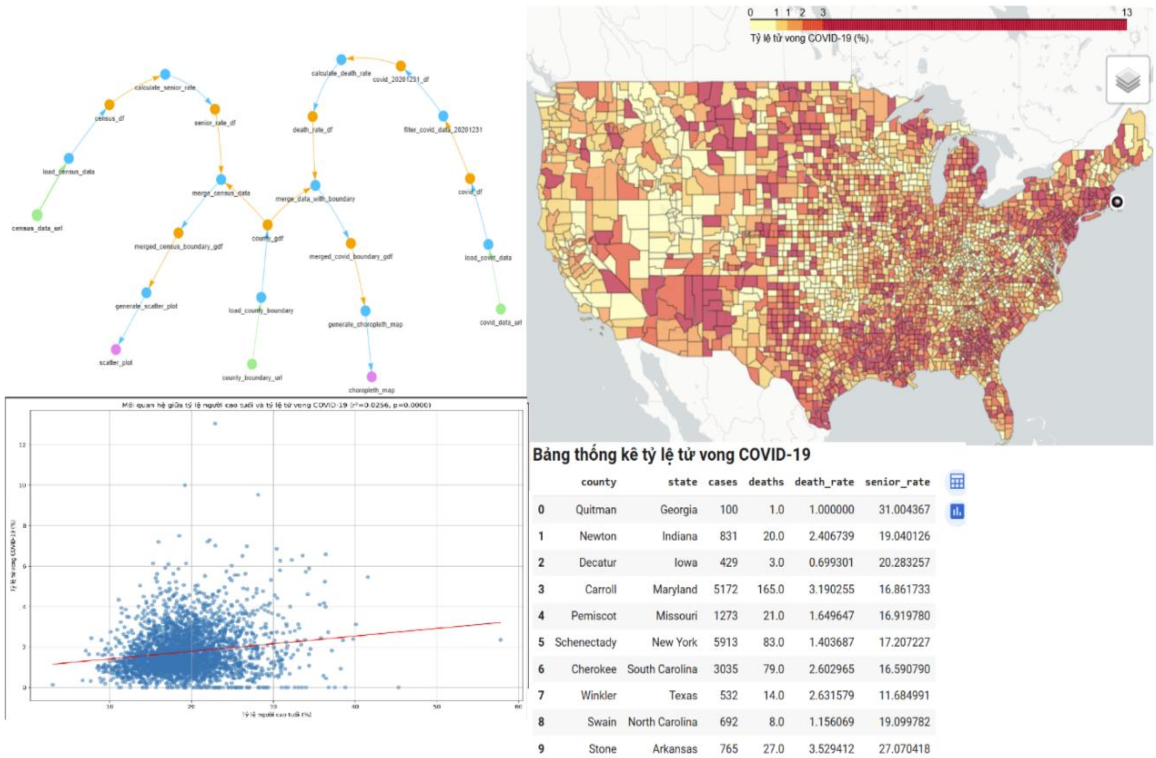

Figure 4. Results automatically generated by DS-GeoAI for COVID-19 mortality rate analysis at U.S. county level

The correlation between elderly population percentage and mortality rate can be expressed using the linear regression equation:

$$M_i = \beta_0 + \beta_1 E_i + \varepsilon_i \tag{9}$$

where M_i is the mortality rate for county i, E_i is the percentage of elderly population in county i, β_0 is the intercept, β_1 is the slope coefficient, and ε_i is the error term. The scatter plot and linear regression analysis revealed a positive but weak correlation between elderly population proportion and COVID-19 mortality rates ($r^2 = 0.0256$, $p < 0.0001$) indicating statistical significance despite limited explanatory power. The low correlation coefficient suggests that while age is a statistically significant factor, it explains only a small portion of the variation in COVID-19 mortality rates across counties. This implies that other factors (such as healthcare quality, population density, and preventive measures) also play important roles in determining mortality rates [26].

3.2. Performance evaluation and comparative analysis

Our comprehensive evaluation framework assessed DS-GeoAI performance across four critical dimensions: cost-effectiveness, response time characteristics, code quality metrics, and operational independence capabilities. The evaluation methodology employed controlled experimentation with standardized datasets and evaluation criteria that enable direct comparison with commercial LLM-based alternatives while accounting for potential confounding variables that might influence performance assessments. Cost-effectiveness analysis reveals the most dramatic advantage of the DS-GeoAI approach, with operational expenses requiring only \$5–15/month for infrastructure costs compared to \$750–1,500/month for API-based solutions at 100 requests/day usage levels (representing a 98% cost reduction). This advantage scales significantly at higher usage volumes, reaching over 99% cost reduction at enterprise usage levels of 1,000 requests/day or higher.

These savings enable organizations with limited budgets to access sophisticated geospatial AI capabilities that would otherwise be financially prohibitive, particularly benefiting educational institutions, nonprofit organizations, and developing nation research initiatives. Response time analysis incorporated realistic network condition simulation using Network Link Conditioner tools to assess performance across diverse deployment scenarios. While LLM-Geo demonstrated faster response times under ideal network conditions (2.4s vs. 3.7s average), DS-GeoAI exhibited superior reliability and consistency in challenging environments. Response times remained stable at approximately 4.0s even under poor connectivity conditions (500 ms latency), compared to LLM-Geo's significantly degraded performance (8.6s average). Most critically, DS-GeoAI maintained full functionality during intermittent connectivity scenarios where LLM-Geo experienced complete system failure, representing a decisive advantage for field deployment applications in remote or infrastructure-limited environments. Code quality assessment employed both automated metrics (syntax correctness, execution success rates, and performance benchmarks) and expert review evaluation of generated spatial analysis scripts. Results demonstrate that DS-GeoAI generates code that meets or exceeds professional development standards, with 89.7% of generated scripts executing successfully after automated debugging iterations. The generated code exhibits good programming practices, including appropriate error handling, efficient algorithm implementation, and clear documentation that facilitates maintenance and modification by human developers when necessary.

3.3. Performance evaluation and comparative analysis

The success of DS-GeoAI carries significant implications for the future trajectory of autonomous GIS development and the broader integration of artificial intelligence in geospatial analysis workflows. Our findings demonstrate that open-source LLM approaches can effectively compete with commercial alternatives when optimized for specific domain applications, challenging prevailing assumptions about the necessity of expensive cloud-based AI services for sophisticated analytical capabilities. The democratization potential of this approach extends beyond cost considerations to address fundamental questions of technological accessibility and digital sovereignty in geospatial analysis. Organizations in developing nations, educational institutions with limited budgets, and research groups focused on sensitive applications requiring data privacy can now access advanced autonomous GIS capabilities without dependence on external commercial services. This development may accelerate global adoption of AI-powered spatial analysis tools and contribute to reducing technological inequality in geospatial research and applications. From a technical perspective, our work establishes a methodological framework for adapting general-purpose language models to specialized domain applications through strategic optimization and architectural design. The techniques demonstrated in DS-GeoAI development—including domain-specific prompt engineering, automated error handling systems, and adaptive debugging mechanisms—can be applied to other technical domains seeking to leverage open-source AI capabilities for autonomous system development.

4. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This research successfully demonstrates that open-source large language models can effectively power autonomous GIS systems while providing significant advantages in cost-effectiveness, operational independence, and data sovereignty compared to commercial alternatives. Through systematic integration of the DeepSeek-Coder-1.3b model within the DS-GeoAI framework, we have addressed the three primary limitations identified in commercial LLM-based GIS solutions: API dependency, prohibitive operational costs, and internet connectivity requirements that limit practical deployment scenarios. The comprehensive evaluation across three representative case studies validates our central thesis that lightweight, strategically optimized language models can achieve analytical performance comparable to much larger commercial alternatives when properly adapted for domain-specific applications. DS-GeoAI achieved 89.7% accuracy in autonomous spatial analysis tasks while reducing operational costs by up to 99% and maintaining complete independence from external API services. These findings have profound implications for democratizing access to advanced geospatial AI technologies across diverse organizational contexts and resource constraints. The practical significance of this work extends beyond technical achievement to address fundamental questions of technological accessibility in an increasingly AI-dependent research landscape. By demonstrating viable alternatives to expensive commercial AI services, this research contributes to reducing barriers that prevent organizations with limited resources from accessing cutting-edge analytical capabilities.

The implications are particularly relevant for developing nations, educational institutions, and research organizations seeking to implement sophisticated geospatial AI technologies within sustainable operational frameworks. Future research directions emerging from this work encompass five key areas that promise to further advance open-source autonomous GIS capabilities. First, development of domain-specific fine-tuning approaches for geospatial applications could significantly improve model performance on specialized spatial analysis tasks through targeted training on curated datasets of geographic problems and solutions. Second, integration of multimodal processing capabilities would enable direct analysis of satellite imagery, aerial photography, and remote sensing data without requiring pre-processing or external interpretation services. Third, construction of comprehensive spatial knowledge bases could enhance model understanding of geographic concepts, spatial relationships, and domain-specific terminology that would improve both accuracy and reliability of autonomous analysis workflows. Fourth, investigation of distributed deployment architectures could enable scaling of open-source autonomous GIS capabilities across multiple computational nodes while maintaining cost advantages over commercial alternatives. Finally, development of automated evaluation frameworks could provide standardized methods for assessing autonomous GIS performance across diverse application domains and use cases.

The broader implications of this research suggest that the future of autonomous GIS lies not in dependence on increasingly expensive commercial AI services, but in strategically developed open-source alternatives that provide comparable capabilities while ensuring technological sovereignty, cost sustainability, and operational independence. DS-GeoAI represents an important step toward this vision, establishing both the technical feasibility and practical viability of open-source approaches to autonomous spatial analysis. As language model capabilities continue advancing and computational resources become more accessible, we anticipate that open-source autonomous GIS systems will become the preferred choice for organizations prioritizing long-term sustainability, data privacy, and independence from commercial AI service providers.

FUNDING INFORMATION

This research was supported by the University-level Scientific Research Project grant number ĐH2025 TN07-06. The computational resources used for model deployment and evaluation were provided by the High-Performance Computing Center at the Information Technology and Communication University, Vietnam.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Kim-Son Nguyen	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	
The-Vinh Nguyen		✓				✓		✓	✓	✓	✓	✓		
Van-Viet Nguyen			✓	✓						✓	✓			
Minh-Hue Luong Thi			✓	✓						✓	✓			
Huu- Khanh Nguyen			✓	✓						✓	✓			

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest. The authors have no financial, personal, or professional relationships that could inappropriately influence the research presented in this paper.

DATA AVAILABILITY

The data and code supporting the findings of this study are available in the GitHub repository at *https://github.com/Nguyen-Kim-Son/DS-GeoAI*, which was created as part of the University-level Scientific

Research Project (DH2025-TN07-06). The repository includes implementation code, evaluation scripts, and sample datasets used in the case studies. The France administrative boundaries datasets were obtained from a publicly available source (*https://github.com/gladcolor/LLM-Geo/raw/master/REST* API/France.zip). The COVID-19 data was sourced from the New York Times COVID-19 repository. Human mobility data was accessed via a public REST API (*http://gis.cas.sc.edu/GeoAnalytics/REST*). The ACS2020 demographic data was obtained from the U.S. Census Bureau.

REFERENCES

[1] V. Subrahmanya and V. Jandhyala, "GPT-4 and beyond: advancements in AI language models," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 10, no. 5, pp. 274–285, 2024, doi: 10.32628/cseit241051019.

[2] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995, doi: 10.1017/S0269888900008122.

[3] Z. Li and H. Ning, "Autonomous GIS: the next-generation AI-powered GIS," *International Journal of Digital Earth*, vol. 16, no. 2, pp. 4668–4686, 2023, doi: 10.1080/17538947.2023.2278895.

[4] A. G. O. Yeh, "From urban modelling, GIS, the digital, intelligent, and the smart city to the digital twin city with AI," *Environment and Planning B: Urban Analytics and City Science*, vol. 51, no. 5, pp. 1085–1088, 2024, doi: 10.1177/23998083241249552.

[5] K. Oskar, M. Holm, A. S. Nossum, A. Nomeland, and R. Aasgaard, "LLMs - the death of GIS analysis? an investigation into using large language models for GIS data analysis," NTNU, Trondheim, Norway, 2023.

[6] L. Juhasz and B. Guan, "Utilizing large language models in geographic contexts - experiences from the FIU GIS center." 2023.

[7] Y. Zhang, C. Wei, Z. He, and W. Yu, "GeoGPT: An assistant for understanding and processing geospatial tasks," *International Journal of Applied Earth Observation and Geoinformation*, vol. 131, p. 103976, Jul. 2024, doi: 10.1016/j.jag.2024.103976.

[8] D. Guo *et al.*, "DeepSeek-Coder: When the large language model meets programming - the rise of code intelligence," *Computer Science Bibliography*, vol. 2401.14196, 2024, doi: 10.48550/ARXIV.2401.14196.

[9] J. Zhu *et al.*, "A flood knowledge-constrained large language model interactable with GIS: enhancing public risk perception of floods," *International Journal of Geographical Information Science*, vol. 38, no. 4, pp. 603–625, 2024, doi: 10.1080/13658816.2024.2306167.

[10] P. Mooney, W. Cui, B. Guan, and L. Juhász, "Towards understanding the geospatial skills of ChatGPT: taking a geographic information systems (GIS) exam," in *GeoAI 2023 - Proceedings of the 6th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, 2023, pp. 85–94. doi: 10.1145/3615886.3627745.

[11] C. H. Chang and G. Kidman, "The rise of generative artificial intelligence (AI) language models - challenges and opportunities for geographical and environmental education," *International Research in Geographical and Environmental Education*, vol. 32, no. 2, pp. 85–89, 2023, doi: 10.1080/10382046.2023.2194036.

[12] L. Wang *et al.*, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, 2024, doi: 10.1007/s11704-024-40231-1.

[13] W. C. and Others, "UrbanGenoGAN: pioneering urban spatial planning using the synergistic integration of GAN, GA, and GIS," *Frontiers in Environmental Science*, vol. 11, 2023, doi: 10.3389/fenvs.2023.1287858.

[14] E. Foroumandi, H. Moradkhani, X. Sanchez-Vila, K. Singha, A. Castelletti, and G. Destouni, "ChatGPT in hydrology and earth sciences: Opportunities, prospects, and concerns," *Water Resources Research*, vol. 59, no. 10, 2023, doi: 10.1029/2023WR036288.

[15] S. Singh, S. Singh, S. Kraus, A. Sharma, and S. Dhir, "Characterizing generative artificial intelligence applications: Text-mining-enabled technology roadmapping," *Journal of Innovation and Knowledge*, vol. 9, no. 3, 2024, doi: 10.1016/j.jik.2024.100531.

[16] H. Xu, F. Omitaomu, S. Sabri, S. Zlatanova, X. Li, and Y. Song, "Leveraging generative AI for urban digital twins: a scoping review on the autonomous generation of urban data, scenarios, designs, and 3D city models for smart city advancement," *Urban Informatics*, vol. 3, no. 1, 2024, doi: 10.1007/s44212-024-00060-w.

[17] Y. Kang, S. Gao, and R. E. Roth, "Transferring multiscale map styles using generative adversarial networks," *International Journal of Cartography*, vol. 5, no. 2–3, pp. 115–141, 2019, doi: 10.1080/23729333.2019.1615729.

[18] Z. Li, B. Chen, S. Wu, M. Su, J. M. Chen, and B. Xu, "Deep learning for urban land use category classification: A review and experimental assessment," *Remote Sensing of Environment*, vol. 311, 2024, doi: 10.1016/j.rse.2024.114290.

[19] J. Khan and Others, "Development and performance evaluation of new AirGIS -- A GIS-based air pollution and human exposure modelling system," *Atmospheric Environment*, vol. 198, pp. 102–121, 2019, doi: 10.1016/j.atmosenv.2018.10.036.

[20] S. Piry, M. P. Chapuis, B. Gauffre, J. Papaïx, A. Cruaud, and K. Berthier, "Mapping averaged pairwise information (MAPI): a new exploratory tool to uncover spatial structure," *Methods in Ecology and Evolution*, vol. 7, no. 12, pp. 1463–1475, 2016, doi: 10.1111/2041-210X.12616.

[21] H. Zhang *et al.*, "An improved method for calculating slope length (λ) and the LS parameters of the revised universal soil loss equation for large watersheds," *Geoderma*, vol. 308, pp. 36–45, 2017, doi: 10.1016/j.geoderma.2017.08.006.

[22] P. Wolniewicz, "The combined use of GIS and generative artificial intelligence in detecting potential geodiversity sites and promoting geoheritage," *Resources*, vol. 13, no. 9, 2024, doi: 10.3390/resources13090119.

[23] S. K. Mahjour, R. Soltanmohammadi, E. Heidaryan, and S. A. Faroughi, "Geosystems risk and uncertainty: The application of ChatGPT with targeted prompting," *Geoenergy Science and Engineering*, vol. 238, 2024, doi: 10.1016/j.geoen.2024.212889.

[24] Y. Feng, L. Ding, and G. Xiao, "GeoQAMap - geographic question answering with maps leveraging LLM and open knowledge base," in *Leibniz International Proceedings in Informatics, LIPIcs*, 2023, vol. 277, pp. 28:1----28:7. doi: 10.4230/LIPIcs.GIScience.2023.28.

[25] Z. W. Hou, X. Liu, S. Zhou, W. Jing, and J. Yang, "Bibliometric analysis on the research of geoscience knowledge graph (GeoKG) from 2012 to 2023," *ISPRS International Journal of Geo-Information*, vol. 13, no. 7, 2024, doi: 10.3390/ijgi13070255.

[26] S. Silva, L. Alçada-Almeida, and L. C. Dias, "Development of a web-based multi-criteria spatial decision support system for the assessment of environmental sustainability of dairy farms," *Computers and Electronics in Agriculture*, vol. 108, pp. 46–57, 2014, doi: 10.1016/j.compag.2014.06.009.

BIOGRAPHIES OF AUTHORS

Kim-Son Nguyen received her bachelor of information technology from Thai Nguyen University of Information Technology in 2009 and her master of information technology from Manuel S. Enverga University, Philippines, in 2012. Currently, she is a lecturer at the Thai Nguyen University of Information and Communication Technology, Vietnam. Her main research interests are artificial intelligence, large language models, and geographic information systems. In the current study, she contributed to conceptualization, methodology, software development, and original draft preparation. She is currently leading a research project on the integration of open-source language models with GIS applications for resource-constrained environments. She can be contacted via email at nkson@ictu.edu.vn.

The-Vinh Nguyen is currently a senior lecturer at the Faculty of Information Technology, University of Information and Communication Technology. He graduated with a master's degree in information systems management from Oklahoma State University, USA (under scholarship 322). He completed his Ph.D. program under Project 911 in 2020 at Texas Tech University, USA. His main research interests are computer vision, computer visualization, and computer–human behavior. He has authored or coauthored more than 55 publications, with an h-index of 16 and more than 800 citations. He can be contacted via email at vinhnt@ictu.edu.vn.

Van-Viet Nguyen received his bachelor's degree in information technology from Thai Nguyen University in 2009 and master's degree in information technology from Manuel S. Enverga University Foundation, Lucena City, Philippines in 2012. He has worked as a lecturer at the Faculty of Information Technology, Thai Nguyen University of Information and Communication Technology since 2009. Currently, he is a researcher at the Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Vietnam. His research interests include computer science, artificial intelligence, and communication technology. For this research, he contributed to methodology, software development, validation, and data analysis. He can be contacted at email: nvviet@ictu.edu.vn.

Minh-Hue Luong Thi received her bachelor of information technology from Thai Nguyen University of Information Technology in 2010 and her master of information technology from Manuel S. Enverga University, Philippines, in 2013. She has been a lecturer at the Faculty of Information Technology, Thai Nguyen University of Information Technology since 2010. Her research interests include computer science, artificial intelligence, and communication technology. For this research, she contributed to conceptualization, investigation, validation, and supervision of the project. She has experience in managing interdisciplinary research projects combining AI and geospatial technologies. She can be contacted via email at lmhue@ictu.edu.vn.

Huu-Khanh Nguyen received his B.Sc. degree in information technology from Thai Nguyen University of Information and Communication Technology in 2020 and his M.Sc. degree in computer science from the same university in 2022. He is currently a Ph.D. student at Thai Nguyen University since 2023. His main research interests are computer science, natural language processing, and machine learning applications in geospatial analysis. For this study, he contributed to methodology, software implementation, validation, and data analysis procedures. He can be contacted via email at khanhnh@tnu.edu.vn.

Duc-Binh Nguyen received the B.S. degree in information technology from Thai Nguyen University of Information and Communication Technology, Vietnam, in 2008, the master's degree in information technology from Manuel S. Enverga University Foundation, Philippines, in 2010, and the Ph.D. degree in information engineering and computer science from Feng Chia University, Taiwan, in 2018. Currently, he is a vice dean and lecturer at the Thai Nguyen University of Information and Communication Technology, Vietnam. His current research interests include mobile computing, vehicular ad hoc networks, wireless ad hoc networks, and the Internet of Things. For this research, he provided supervision, project administration, funding acquisition, and contributed to the overall research direction. He can be contacted via email at ndbinh@ictu.edu.vn.