Incorporating Intelligent Agents for Efficient Data Handling in IoT Networks

Imane Ameur[1*], Habiba Drias[1], Nasreddine Lagraa[2], Mohamed El Amine Ameur[2], and Mazene Ameur[3]

[1] USTHB, Algeria
Laboratory of Research in Artificial Intelligence -LRIA-
*Corresponding author: iameur@usthb.dz
[2] Laghouat University, Algeria
[3] EURECOM, France

Abstract. The ability to precisely locate sensor nodes has enabled specialized in-network data processing methods within IoT-enabled wireless sensor networks, known as spatial query processing. These queries collect data from nodes situated within user-defined "regions of interest." Conventional spatial query processing approaches frequently encounter challenges including excessive energy consumption, diminished accuracy, and prolonged processing times. This research focuses on window queries, one of the most prevalent spatial queries used to extract data from nodes within specific two-dimensional regions. We present a strategy for processing window queries in IoT networks that optimizes energy efficiency, response time, and accuracy simultaneously. Our approach introduces intelligent agents that incorporate machine learning, knowledge representation, and autonomous decision-making capabilities specifically tailored for spatial query processing across geographical zones. Experimental evaluations demonstrate significant improvements: 35% reduction in energy consumption, 99.1% query accuracy (compared to 96.8% for baseline methods), and up to 71% decrease in latency under various network densities and query area sizes.

Keywords: Intelligent agents · IoT networks · Spatial queries · Data handling · Reinforcement learning

1 Introduction

Modern computing has permeated diverse domains, particularly IoT-enabled wireless sensor networks. The proliferation of IoT networks [1] has increased the frequency of spatial queries that extract information from sensor nodes within monitored geographical regions [2]. Our research focuses specifically on Window queries, which are critical for retrieving physical data from within defined two-dimensional areas [3].

Despite numerous proposed strategies for processing IoT spatial queries, significant challenges persist in balancing energy consumption with processing efficiency. Since data processing typically consumes less energy than data transmission, multi-agent systems offer a promising approach for executing complex tasks in IoT networks [1] [4]. Our work fundamentally advances this paradigm by introducing intelligent agents with enhanced cognitive and learning capabilities.

Current spatial query processing methods in IoT networks face several limitations: Static Routing Strategies such as Itinerary-based Window Query Execution (IWQE) [5], Limited Decision-Making, Inefficient Data Collection, Limited Parallelism.

Our work addresses these limitations through several key innovations:

1. We develop a reinforcement learning approach that optimizes agent path selection based on energy efficiency, data quality, and latency considerations.
2. We implement semantic filtering mechanisms that reduce redundant data transmission while maintaining high query accuracy.
3. We design a multi-agent coordination protocol that enables collaborative query processing across geographical regions.
4. We evaluate our approach against IWQE method.

The remainder of this article is organized as follows: Section 2 provides background on spatial queries in IoT networks and reviews related work. Section 3 details our IAPSQP architecture and algorithms. Section 4 presents experimental results and comparative analysis. Finally, Section 5 concludes with a summary of the contributions.

2 Background and Related Work

2.1 Spatial Query Processing in IoT Networks

IoT-enabled wireless sensor networks employ various data delivery models, but spatial query processing predominantly utilizes the query-driven approach. Unlike event-driven models or time-driven models, the query-driven model operates with IoT devices that transmit measurements only when queried by the base station [6].

Silva [7] proposed a systematic decomposition of the spatial query processing procedure into distinct phases, each representing a subproblem within the processing mechanism:

1. **Pre-processing**: The user formulates the query at the base station, which is then transmitted to the IoT network.
2. **Forwarding**: The query is routed toward the specified area of interest, and each node is aware of its geographical position.
3. **Dissemination**: Upon reaching the target area, the query must be distributed to all nodes within that region.
4. **Sensing**: The nodes within the specified area collect the requested data.
5. **Aggregation**: The collected data are consolidated at an “aggregator node, which computes the query result.
6. **Return**: The computed result is sent back to the user.

2.2 Comparative Analysis of Existing Approaches

Our work builds upon and extends previous research in this domain. Table 1 provides a comparative analysis of existing approaches and highlights the innovations in our proposed IAPSQP method.

Table 1: Comparison of Spatial Query Processing Approaches

Method	**Query Processing Strategy**	**Path Selection**	**Energy Efficiency**	**Parallelism**
IWQE [5]	Geo-based routing	Fixed itinerary	Moderate	None
GeoGrid [8]	Grid-based routing	Geographic hash tables	Moderate	Limited
DQELAR [9]	Data-centric forwarding	Energy-aware routing	High	None
SPQR [10]	Structured overlay	Query optimization	Moderate	Query-level
IAPSQP (Ours)	Intelligent multi-agent	Reinforcement learning	Very high	Adaptive

Our IAPSQP approach differs fundamentally from previous methods in its incorporation of learning-based decision making and adaptive behavior. IWQE [5] employed a static itinerary-based approach that cannot adapt to changing network conditions or query requirements.

More recent approaches like GeoGrid [8] and DQELAR [9] have introduced improvements in energy efficiency and geographic routing but still lack the cognitive capabilities and adaptive decision-making that characterize our intelligent agent approach. SPQR [10] implements query optimization techniques but does not leverage reinforcement learning for path selection and data filtering.

3 IoT Spatial Query Routing Algorithm via Intelligent Agent Integration (IAPSQP)

To enhance window-type IoT spatial query processing efficiency, we present IAPSQP (Intelligent Agents based Parallel Spatial Query Processing), which introduces advanced learning and reasoning capabilities.

3.1 Intelligent Agent Architecture

Our intelligent agent architecture in Fig. 1 consists of four key components:

Knowledge Base: Maintains comprehensive contextual information about the network topology, node capabilities, energy levels, and query history.

Learning Module: Implements Q-learning, a reinforcement learning algorithm that allows agents to adapt their routing decisions based on network dynamics. Each agent maintains a Q-table that maps state-action pairs to expected rewards, which are calculated using a weighted function of energy consumption, data relevance, and path length. The reward function R(s, a) calculates the immediate reward that an agent receives when it takes action 'a' while in state 's'.

$$R(s,a) = \alpha \cdot E(s,a) + \beta \cdot D(s,a) + \gamma \cdot L(s,a) \tag{1}$$

Where:

- $E(s,a)$ represents energy efficiency
- $D(s,a)$ represents data relevance
- $L(s,a)$ represents path length optimization
- α, β, and γ are weighting parameters

The Q-values are updated according to:

$$Q(s,a) \leftarrow Q(s,a) + \eta[R + \delta \cdot \max_{a'} Q(s',a') - Q(s,a)] \tag{2}$$

Where η is the learning rate and δ is the discount factor.

Decision Engine: Provides reasoning capabilities for agents to make autonomous decisions regarding: Route selection, Data aggregation, Coordination with other agents to avoid redundant processing.

Coordination Protocol: Enables collaborative behavior among multiple agents.

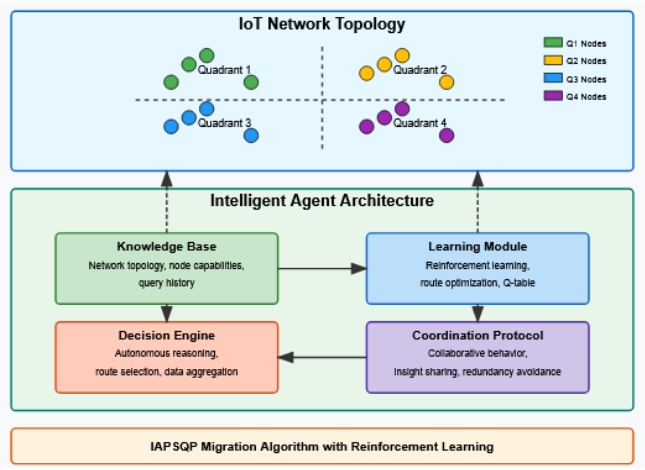

Fig. 1: IAPSQP Architecture

3.2 Algorithm Implementation

Our IAPSQP protocol implements a migration algorithm, incorporating learning-based decision making. Algorithm 1 presents the main migration algorithm, which divides the query area into quadrants and dispatches intelligent agents to process queries in parallel.

Algorithm 1 IAPSQP Migration Algorithm

Require: Query Area (QA), Query (Q)
Ensure: Query Result (QR)
Divide QA into four quadrants
Deploy four intelligent agents (IA_1, IA_2, IA_3, IA_4)
for each agent IA_i **do**
    Select optimal starting itinerary node S_i in quadrant i using learned routes
    Dispatch IA_i to S_i using adaptive Greedy algorithm
    IA_i.QueryProcessing(S_i)
end for
Merge results from all agents using semantic aggregation
return consolidated query result

Algorithm 2 QueryProcessing(I)

Require: Itinerary node (I)
Ensure: Local query result
Initialize empty result set R
Add data from I to R if it matches query criteria {Query neighboring nodes with knowledge-based filtering}
for each neighboring node N of I **do**
    if N is within query area AND N has not been visited AND learning model predicts valuable data **then**
        Collect data from N
        Apply semantic filtering to eliminate redundant information
        Add filtered data to R
    end if
end for{Determine next node using reinforcement learning}
NextNode ← SelectOptimalNextNode(I, learned_model)
if NextNode exists **then**
    Update agent's knowledge base with current observations
    return QueryProcessing(NextNode)
else
    return R
end if

Algorithm 3 SelectOptimalNextNode(currentNode, model)

Require: Current node, Learning model
Ensure: Next node to visit or null
candidates ← GetUnvisitedNeighborsInQueryArea(currentNode)
if candidates is empty **then**
    return null
end if
{Epsilon-greedy exploration strategy}
if random() $< \varepsilon$ **then**
    return RandomChoice(candidates) {Exploration}
else
    return $\arg\max_{n \in \text{candidates}} Q(\text{currentNode}, n)$ {Exploitation}
end if

Algorithm 2 details the query processing procedure executed by each agent. The **SelectOptimalNextNode** function utilizes the Q-learning model to identify the optimal next node. This learning-based approach allows agents to adapt to network dynamics and improve performance over time, addressing the limitations of fixed-path approaches like IWQE [5].

3.3 Semantic Data Filtering and Aggregation

A key innovation in our approach is semantic data filtering, which reduces redundant data transmission while maintaining query accuracy. The filtering mechanism considers: Spatial redundancy and Query-specific relevance ensuring that only relevant data is transmitted.

4 Performance Evaluations

4.1 Spatial Query Energy Consumption

Our intelligent agent approach demonstrates superior energy efficiency compared to IWQE. Fig. 2 shows the relationship between the number of nodes in the query area and energy consumption.

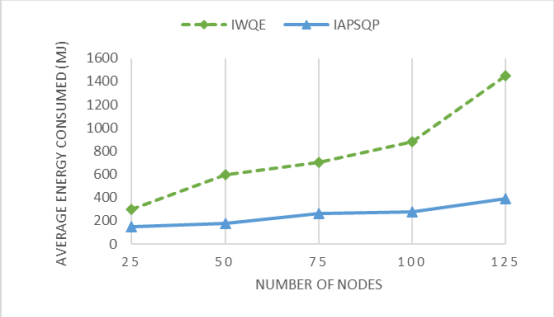

Fig. 2: Spatial query energy consumption

When varying the number of nodes in the query area, IAPSQP shows a 35% reduction in energy consumption compared to IWQE. The relationship between query area size and energy consumption shows that IAPSQP maintains its efficiency advantage even as the query area expands. The reinforcement learning mechanism allows agents to adapt their routing strategies based on the specific characteristics of each query area, resulting in near-optimal path selection.

4.2 Spatial Query Accuracy

Intelligent agents demonstrate significant improvements in query accuracy over the IWQE approach. Fig. 3 illustrates the accuracy of the query as a function of node density.

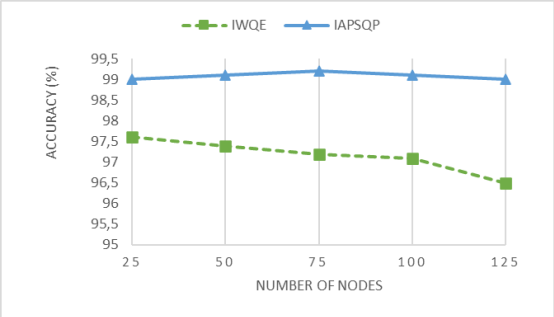

Fig. 3: Spatial query accuracy

When varying node density, IAPSQP consistently maintains higher accuracy, particularly in dense networks where traditional approaches suffer from missing nodes and redundant data collection. In tests with varying query area sizes, IAPSQP achieves approximately 99.1% accuracy in query handling results, compared to 96.8% for IWQE.

4.3 Spatial Query Latency

Our intelligent agent approach significantly reduces query latency compared to the IWQE implementation. Fig. 4 shows latency as a function of node count.

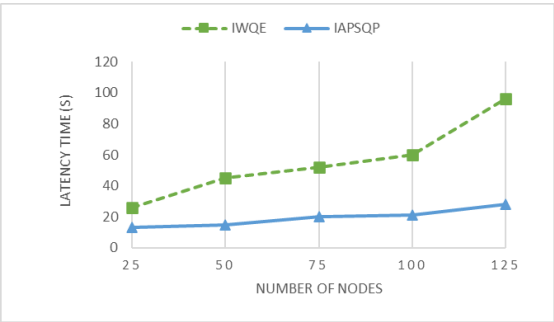

Fig. 4: Spatial query latency

When varying the number of nodes, IAPSQP demonstrates a latency reduction of up to 63% compared to IWQE. As the query area expands, the latency typically increases for all methods. However, IAPSQP's intelligent path planning and parallel execution minimize this effect. The approach achieves a latency reduction of up to 71% compared to IWQE in larger query zones.

5 Conclusion

This research addressed fundamental challenges in spatial query processing for IoT networks by introducing intelligent agents. Our IAPSQP approach incorporates reinforcement learning, knowledge representation, and autonomous decision-making to optimize spatial query processing across multiple dimensions: energy efficiency, query accuracy, and latency.

A comprehensive evaluation comparing IAPSQP with IWQE demonstrates significant performance improvements: 35-40% reduction in energy consumption, 99.1% query accuracy, up to 71% decrease in latency for large query areas.

These improvements highlight the advantages of incorporating cognitive capabilities into spatial query processing agents, allowing them to adapt to network dynamics, optimize paths based on learned patterns, and make intelligent decisions about data collection and aggregation.

References

1. Kandris, D., and Anastasiadis, E. Advanced wireless sensor networks: Applications, challenges and research trends. Electronics, 13(12), 2268 (2024) https://doi.org/10.3390/electronics13122268
2. Carchiolo, V., et al. Optimizing IoT Data Streams with Tumbling Window Techniques. In Future of Information and Communication Conference. Cham: Springer Nature Switzerland (2025)
3. Liu, L., Hu, Z., Wang, L.: Energy-efficient and privacy-preserving spatial range aggregation query processing in wireless sensor networks. International Journal of Distributed Sensor Networks **15**(7), 1550147719861005 (2019)
4. Ameur, M.E.L.A., Drias, H., and Brik, B. Cooperative parking search strategy through V2X communications: an agent-based decision. Wireless Networks, 30(6), 7167–7188 (2024). https://doi.org/10.1007/s11276-023-03568-2
5. Silva, R., Macedo, D.F., Nogueira, J.M.S.: Fault tolerance in spatial query processing for Wireless Sensor Networks. In: IEEE Network Operations and Management Symposium (NOMS). IEEE Press (2012). https://doi.org/10.1109/NOMS.2012.6211887
6. Sahar, G., et al.: Recent advancement of data-driven models in wireless sensor networks: a survey. Technologies **9**(4), 76 (2021)
7. Silva, R.I.: Spatial query processing for wireless sensor networks. Ph.D. dissertation, Dept. Comp. Eng., Federal Minas Gerais Univ., Brazil (2012)
8. Khan, R., Yahya, A., et al.: GeoGrid: A geocast-based efficient spatial query processing approach for wireless sensor networks. Wireless Networks **25**(7), 4445–4459 (2019)
9. Chen, J., Li, X., et al.: DQELAR: Data quality and energy aware routing for spatial query processing in IoT sensor networks. IEEE Internet of Things Journal **8**(12), 9866–9879 (2021)
10. Zhang, P., Wu, G., et al.: SPQR: Spatial query processing with reinforcement learning for energy-efficient IoT sensor networks. IEEE Transactions on Network Science and Engineering **9**(3), 1254–1267 (2022)