AgroSAGE: A Modular Ontology-Driven Framework for Real-Time Sensor Annotation in Smart Agriculture

Mostafha Alwbaidy[1]*

[1]Department of Computers Techniques Engineering, College of Technical Engineering, The Islamic University, Najaf, Iraq

Abstract. Global food security issues are growing due to increasing population, farmland decline, and climate instability, raising the need to transition to smart agriculture by utilizing the Internet of Things (IoT) and Artificial Intelligence (AI) for efficient production. Existing IoT frameworks lack domain-specific ontologies, and the processing of high-dimensional sensor data increases cost and latency. AgroSAGE (Smart Annotation and Graph-based Embedding for Agriculture) is proposed. First, agricultural ontologies are constructed using Natural Language Processing (NLP)-based entity extraction and Graph Neural Networks (GNN), capturing crop types, phenology stages, and agronomic actions. Second, a semantic- and ontology-based IoT framework enables protocol reconciliation and semantic mapping across heterogeneous devices. Finally, Dynamic Sparse Principal Component Analysis (D-SPCA) was applied to reduce the number of sensors used while preserving variance and interpretability. The ontology is modularized into crops, sensors, phenology, and action components for lightweight deployment. Real-time annotation was performed at the edges using compressed GNN embeddings and semantic rules. The validation protocol simulated 100–500 devices using MQTT, CoAP, and HTTP. AgroSAGE achieves global sparsity with fewer sensors while maintaining prediction accuracy effectively, resulting in 98% interoperability success, 66% latency reduction, and balanced resource utilization.

1 Introduction

Agriculture has an essential impact on the prosperity and food security of any country and plays a significant role in economic growth [1]. The cultivation and management of agricultural crops have become challenging because of fluctuations in environmental conditions, soil factors, and water quality around the world [2]. Traditional agricultural practices, which are based on soil mineral consumption and overuse of water, are factors that lead to environmental degradation. Agriculture uses a large amount of chemicals, such as synthetic fertilizers and pesticides, which have negative effects on human health and the ecosystem, contributing to soil and water pollution [3]. The agricultural industry has changed considerably to address the demand created by the increasing population of the world, sustainability issues, and changing climate [4]. The global population is predicted to reach 8.5 and 9.7 billion by 2030 and 2050, respectively, which requires an immediate introduction of sustainable food production solutions and climate change mitigation measures [5]. New technologies and solutions are also being introduced to the world continuously to address the global needs of humans [6]. The emergence of new technologies for the proliferation of agricultural and farm activities is cost-effective.

Advanced technologies, such as the Internet of Things (IoT), are employed in smart farming applications that work automatically [7]. The Internet of Things (IoT) redefines agriculture by transforming traditional farming into a dynamic, data-driven, and sustainable system tailored to meet growing global food demand [8] [9]. Furthermore, the utilization of edge servers to integrate and process sensed data from end devices can alleviate the burden of data processing in the cloud. Deploying webcams in farmland provides overall crop growth, whereas mobile robots can perform individualized monitoring and management. By combining these two image acquisition methods, a real-time monitoring and early warning system based on edge intelligence can be established. This system can enable refined agricultural management, improve crop growth efficiency and yield, and reduce resource waste [10]. Nevertheless, these advancements, challenges remain in semantic annotation, interoperability, and real-time automation. Existing Internet of Things (IoT) frameworks lack domain-specific ontologies, whereas high-dimensional sensor data increases cost and latency. To address these limitations, AgroSAGE (Smart Annotation and Graph-based Embedding for Agriculture) was proposed, and its contributions are as follows:

- Agricultural ontologies are constructed using Natural Language Processing (NLP)-based entity extraction and Graph Neural Networks (GNN), capturing crop types, phenology stages, and agronomic actions.

* Corresponding author: mostafha.alwbaidy.iu@gmail.com

- A semantic- and ontology-based IoT framework enables protocol reconciliation and semantic mapping across heterogeneous devices.
- Dynamic Sparse Principal Component Analysis (D-SPCA) was applied to reduce the number of sensors used while preserving variance and interpretability.
- The ontology in AgroSAGE is modularized into crops, sensors, phenology, and action components for lightweight deployment. Real-time annotation was performed at the edges using compressed GNN embeddings and semantic rules. The validation protocol simulated 100–500 devices using MQTT, CoAP, and HTTP.

The remainder of this paper is organized as follows. Section 2 presents a literature survey, Section 3 explains the proposed methodology, Section 4 presents the results and discussion, and Section 5 concludes the paper with future directions.

2 Literature survey

This section provides a literature survey on recent advancements in smart agriculture technologies and explains their methodology, strengths, and limitations.

Nawaz et al. [11] implemented a low-cost, open-source IoT architecture based on a three-layer system, sensing, network, and application using Arduino Nano 33 BLE and Raspberry Pi 3 B + with BLE-MQTT communication. Smart agriculture was achieved through real-time monitoring of environmental and soil parameters in a greenhouse, enabling irrigation decisions through derived indicators, such as Growing Degree Days and Vapor Pressure Deficit. The modular design allows flexible deployment across small farms with minimal infrastructure. However, reliance on BLE and MQTT introduced susceptibility to data transmission delays in areas with unstable connectivity.

Adamo et al. [12] presented an IoT-based micro-services framework using an AI-powered autonomous agent modeled through Integer Linear Programming (ILP) to optimize irrigation and fertigation decisions. Smart agriculture has been achieved by integrating real-time sensor data on soil nutrients and moisture with modular cloud services that dynamically adjust water and fertilizer delivery per crop-specific threshold. The architecture enables fine-grained control and flexible service orchestration across heterogeneous farm setups, although its reliance on cloud connectivity introduces latency risks in low-bandwidth rural environments.

Akbari et al. [13] employed an AoI-aware energy-efficient Service Function Chaining (SFC) method using asynchronous federated learning across UAV-assisted IoT nodes in smart agriculture. Real-time sensing and model updates are coordinated through distributed learning without central aggregation, enabling timely decisions on irrigation and crop health. The architecture enhances data freshness and reduces redundant transmissions, although its reliance on UAV mobility prediction introduces scheduling complexity under uncertain flight conditions.

Shahab et al. [14] implemented an IoT-based smart soil monitoring system using an 8-in-1 sensor integrated with ESP8266 and a cloud platform to measure the temperature, moisture, salinity, EC, pH, and NPK levels in rice fields. Real-time data were transmitted wirelessly for analysis and visualization, enabling farmers to optimize irrigation and fertilization through an AI-driven mobile application. The system allowed precise nutrient tracking and remote decision-making, although its dependence on mobile Internet connectivity introduced reliability concerns in low-signal rural zones.

Aldossary et al. [15] presented a hybrid IoT-enabled framework using machine learning models such as MLP, Naïve Bayes, and SVM for anomaly detection in dry bean datasets and deep learning models such as MobileNetV2, VGG16, and InceptionV3 for soil type classification. Smart agriculture was achieved by deploying sensor-equipped IoT nodes to collect real-time environmental and crop data, which were processed in the cloud for autonomous decision making. The architecture supported the modular integration of classifiers and remote monitoring, although its dependence on centralized cloud infrastructure introduced latency risks in low-connectivity agricultural zones.

Although recent smart agriculture systems offer embedded sensing and rule-based control, they lack semantic interoperability, dynamic sensor optimization, and scalable annotation. The proposed AgroSAGE framework addresses these gaps by integrating NLP-GNN-based ontology construction for modular annotation, D-SPCA for relevance-preserving dimensionality reduction, and semantic middleware that harmonizes MQTT, CoAP, and HTTP into a unified annotated stream. A detailed explanation of this is provided in the following section.

3 Proposed methodology

This section presents a hybrid framework that integrates agricultural ontology construction, semantic IoT interoperability, and dimensionality reduction for real-time sensor annotation in smart agriculture.

The proposed AgroSAGE framework combines three components, as shown in Figure 1: (1) domain-specific ontology generation using NLP and graph neural networks, (2) semantic mapping and protocol reconciliation for heterogeneous IoT devices, and (3) dynamic sparse principal component analysis to reduce sensor dimensionality while preserving interpretability and variance. Each component is designed to address specific limitations in existing smart agriculture systems, including the lack of domain-specific ontologies, high-dimensional sensor data, and protocol-level heterogeneity. The framework is structured to ensure semantic consistency, modular deployment, and computational efficiency across edge and cloud environments.

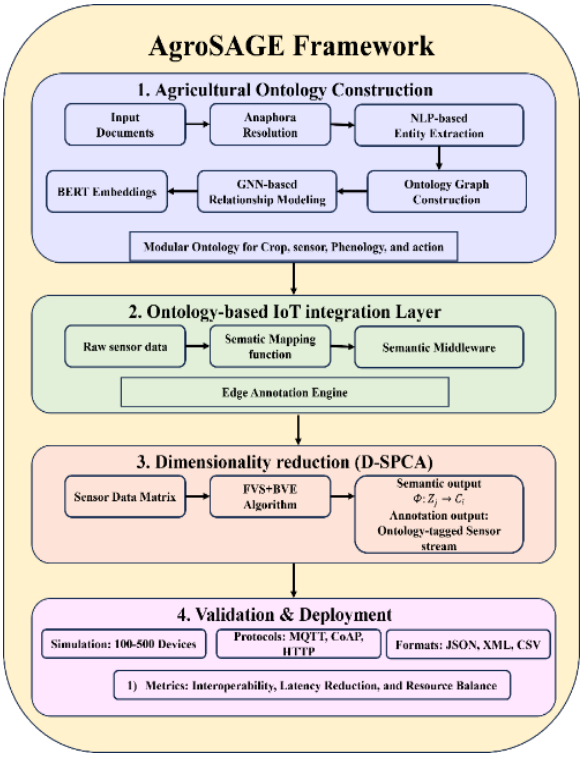

Fig. 1. The architecture of AgroSAGE framework.

3.1 Agricultural ontology construction using NLP and GNN

The aim of the proposed agricultural ontology construction's aim is to develop an ontology for the agricultural domain from input documents. To develop an effective ontology, agricultural objects and the connection relations between entities must be properly identified and extracted. The framework follows a multistage pipeline comprising semantic preprocessing, domain-specific term extraction, relationship identification, and graph-based embedding.

3.1.1 Semantic preprocessing, anaphora resolution, domain-specific term extraction, and relationship identification using hearst patterns

Handling agricultural text documents requires addressing the frequent presence of anaphors to preserve their semantic integrity. Therefore, the input data process for a nominal anaphora resolution phase employs an ontology-driven methodology. The NP identifier function extracts noun phrases from annotated sentences, and the mapper function substitutes anaphors with antecedents using a similarity matrix computed through cosine similarity. Semantic similarity is calculated, where rows represent anaphoric sentences and columns represent antecedents. The row-wise maximum is used to match and replace the anaphors, ensuring semantic completeness before term extraction. This preprocessing step ensures that all subsequent NLP operations operate on semantically complete text, preserving referential integrity across agronomic descriptions. The resolved documents serve as the input for term extraction and relationship modeling.

The candidate agricultural terms were captured using linguistic filters and rule-based mechanisms. This process includes stop word removal, n-gram modeling, and a concatenation filter to generate a candidate list. Term selection is performed by validating a specialized agricultural vocabulary organized from multiple sources. This ensures that only relevant entities such as crop types, phenology stages, and agronomic actions are retained for ontology construction. The resultant set of words after stop-word removal and the n-gram model is the candidate list, which is then refined further by another concatenation filter. The extracted terms were finally checked with the vocabulary arranged with the use of various sources. This step ensured lexical precision and domain relevance.

Once domain-specific terms are extracted, the relationships between them are identified using Hearst patterns. These are categorized as follows:

Hearst Pattern 1: Detects hyponyms using phrases like "such as," "including," etc.

Hearst Pattern 2: Identifies coordination-based relations.

Heart Pattern 3: Extracts specific hierarchical relations, though less frequently found in text.

These patterns enable the diagnosis of standard domain relationships and form the semantic edges of an ontology graph. Hearst Pattern 3 is the specific pattern and the less frequent of which this pattern is observed in the text. However, it is most probable to extract the correct hypernym relations. This rule-based relationship extraction complements the term identification process and provides a structured mechanism for linking the entities within the agricultural domain.

3.1.2 Embedding-based entity recognition, graph-based relationship modeling with GNN, and ontology graph construction

To enhance semantic richness, a pretrained BERT model with NLP techniques is employed to generate word embeddings. These embeddings capture latent syntactic and semantic features that are further processed using regular expressions and positional features to detect nuanced relationships. This embedding-based approach complements rule-based Hearst patterns, allowing the identification of implicit or context-dependent associations that may not conform to fixed linguistic templates. The BERT model enables contextualized representation of agricultural terms, improves the accuracy of relationship detection, and supports downstream graph construction.

The extracted embeddings is processed by an unsupervised GNN network using PyTorch and PyTorch geometry. The model has two linear layers, in which the node features are transformed into hidden and output dimensions. The input and output dimensions match and the model replicates the input features in an autoencoder. The GNN is applied with the Mean Squared Error (MSE) as the loss function, Adam optimizer, and Mish activation function, achieving the best result. Hyperparameter tuning confirms that using a learning rate of 0.01 is effective in minimizing MSE. We also extract from the relationships using regular

expressions and represent them in the form of a graph with entities as nodes and relationships as edges. The MSE loss is used to provide the performance loss of the GNN. To minimize MSE, the hyperparameters in the GNN model is tuned. A learning rate of 0.01 performs better with the Mish activation function and Adam optimizer.

The final ontology graph integrates all the extracted entities and relationships. It is modularized into four components such as Crop Ontology, Sensor Ontology, Phenology Ontology and Action Ontology This modular structure supports lightweight deployment and enables semantic annotation at the edge using compressed GNN embeddings and rule-based mappings. The created ontology contained words as nodes and relations as edges. The objects are graph nodes and the connections is graph edges.

3.2 Semantic and ontology-based IoT framework

To enable semantic interoperability across heterogeneous IoT devices, the proposed framework integrated ontologies and middleware for protocol reconciliation and structured data transformation. The architecture is organized into three primary layers: Semantic & Ontology Layer, Cloud/Edge Layer, and IoT Devices and Communication Layer.

3.2.1 Semantic, ontology layer and IoT devices and communication layer

This provides semantic interoperability by transforming raw IoT data (Draw) into structured semantic information ($D_{Semantic}$), leveraging the rules, as shown in Equation (1):

$$D_{Semantic} = F(D_{raw}) \tag{1}$$

The ontology Model (O_{IoT}), which defines IoT concepts, relationships, and hierarchies, is represented in Equation (2):

$$O_{\mathrm{IoT}} = \{C, R, H\} \tag{2}$$

Where C denotes concepts such as actuators and sensors, R denotes relationships, and H denotes hierarchical links. The Semantic & Ontology Layer guarantee semantic consistency across devices and support rule-based annotation by using the modular ontology generated in Section 1.

The Cloud/Edge layer enables distributed processing such as local data, and the edge component allows local processing of data I_{edge} which reduces latency and supports real-time processing. The Cloud Component handles large-scale analytics and long-term storage, as shown in Equation (3).

$$I_{edge} = P(D_{semantic}) \tag{3}$$

The edge component supports real-time annotation using compressed GNN embeddings, whereas the cloud performs latency-prone analytical workflow across distributed nodes. This layer manages the device-level communication and protocol handling. A unified format (P_{unified}) is derived from varied protocol streams (P_i) through semantic middleware mapping, as shown in Equation (4):

$$M_{proto} : P_i \rightarrow P_{\mathrm{unified}} \tag{4}$$

This mapping ensures seamless interaction across devices by using MQTT, HTTP, and CoAP. This framework manages semantics and ontologies, allowing IoT devices to communicate, interpret, and understand data semantically. The design facilitates smooth information processing, interpretation, and interaction with devices.

3.3 Dimensionality reduction using D-SPCA

The proposed framework integrates a dimensionality reduction module based on D-SPCA to address the challenges of high-dimensional sensor data in smart agriculture. This component is designed to extract interpretable features while preserving the maximum variance from the original dataset, enabling efficient downstream processing and semantic annotation.

3.3.1 Motivation and sparse representation

Traditional Principal Component Analysis (PCA) methods, which are effective in variance preservation, often result in dense loading vectors that hinder interpretability. In contrast, D-SPCA introduces sparsity constraints to ensure that each principal component (PC) is composed of a limited number of original variables. This sparsity facilitates a domain-specific interpretation and supports modular deployment in edge environments.

The sparse loading vector a for the jth principal component is computed by addressing the following constrained optimization formulation, as shown in Equation (5):

$$\begin{aligned} max \quad & a_j^T \Sigma_j a_j \\ s.t. \quad & a_j^T a_j = 1 \\ & \left|\left|a_j\right|\right|_0 \leq K_j \end{aligned} \tag{5}$$

Where Σ represents the covariance matrix of the input data, $\left|\left|a_j\right|\right|_0$ denotes the number of nonzero elements a, enforcing sparsity, and K is the sparsity threshold that controls the number of selected variables. Therefore, this formulation ensures that the selected components retain a high variance while being composed of a minimal subset of features.

3.3.2 Algorithmic framework

The D-SPCA algorithm is aimed at producing global sparsity in the first few principal components. The use of PCs ensures that the extracted features interact with a

limited number of sensors. This method can be viewed as having two major components: Forward Variable Selection (FVS) and Backward Variable Selection (BVS).

3.3.2.1 Forward Variable Selection (FVS)

The FVS procedure is responsible for identifying and adding the variables that contribute the most to the explained variance (EV) of each PC. Given a candidate variable set CV, number of variables used in the j^{th} PC k_j, and current set of used variables U_j, the algorithm evaluates each candidate by temporarily adding it to U_j, forming a new set $l = U_j \cup \{i\}$. For each set l, the subcovariance matrix $\Sigma(l, l)$ is extracted, and the largest eigenvalue is computed using Equation (6):

$$eigvals, eigvecs = eig\big(\Sigma(l, l)\big) \tag{6}$$

The eigenvalue corresponding to the maximum variance is recorded in a variance set V_j. After evaluating all the candidates, the variable that yields the highest EV is selected and added to U_j is incremented by one. This process continues until the number of selected variables reaches the preset threshold, K_j.

3.3.2.2 Backward Variable Elimination (BVE)

Once the number of variables in the j^{th} PC exceeds two, the BVE procedure is activated to refine the sparsity pattern. The goal is to remove variables that do not significantly contribute to the EV. For each variable i in U_j, a new $set\ l = U_j \setminus \{i\}$ is formed. The subcovariance matrix $\Sigma\ (l, l)$ is computed, and the largest eigenvalue is determined. These values are stored in a new variance set, V_j. If the maximum EV in V_j exceeds the current EV Var_j, the corresponding variable is removed from U_j, and K_j is decremented by one. This iterative pruning process ensures that only the most informative variables are retained.

The combined FVS-BVE strategy ensures that the final loading vectors are both sparse and variance-optimal. The algorithm dynamically adjusts the sparsity level based on input data characteristics, making it suitable for heterogeneous sensor streams in agricultural deployments.

3.4 Feature matrix construction

Let $X \in R^{n \times p}$ be an input data matrix with n samples and p features. The D-SPCA algorithm produces a reduced feature matrix $Z \in R^{n \times k}$, where $k \ll p$. Each column of Z corresponds to a sparse principal component, as shown in Equation (7):

$$Z = XA \tag{7}$$

Where $A \in R^{p \times k}$ is the matrix of the sparse loading vectors. This transformation retains the essential structure of the data while reducing the dimensionality and computational overhead.

3.4.1 Dynamic adaptation and edge suitability

Unlike static sparse PCA methods, D-SPCA incorporates dynamic adaptation mechanisms that adjust the sparsity threshold, KK, based on real-time data variance and edge resource constraints. This makes the method particularly suitable for deployment in edge-based agricultural systems, where computational resources are limited, and interpretability is critical. The algorithm ensures that the retained variance exceeds a pre-defined threshold (e.g., 90%), thereby balancing dimensionality reduction with information preservation.

3.5 Integration with ontology-based annotation

The reduced feature matrix Z is used for semantic annotation within the ontology-based framework. Each sparse component is mapped to a domain-specific concept (e.g., soil moisture, leaf chlorophyll, and irrigation status) using rule-based and embedding-based techniques. This mapping enhances the interpretability of the sensor data and supports automated decision-making.

The integration is performed through a semantic mapping function Φ, as represented in Equation (8):

$$\Phi: Z_j \rightarrow C_i \tag{8}$$

Where Z_j is the j^{th} sparse component, and C_i is the corresponding concept in the agricultural ontology. This linkage enables real-time annotation and reasoning over sensor data streams.

4 Results and discussion

This section describes the performance of the AgroSAGE framework by evaluating various state-of-the-art systems. The results validate the accuracy, speed, and performance of the different protocols.

4.1 Software and experimental setup

AgroSAGE was installed on a 64-bit Ubuntu Server 22.04 LTS using a GeForce RTX A6000 with (48GB VRAM) and 256GB RAM. GNN training and entity recognition based on BERT were accelerated through Python 3.10, CUDA v12.1 and cuDNN. PyTorch 2.0.1 and PyTorch Geometric 2.3.0 (core modules), RDFlib 6.3.2, and PySHACL 1.7.0 were used. The interoperability between protocols was done through Dockerized MQTT, CoAP, and HTTP middleware. Semantic interoperability, reduction of latency, and efficient utilization of resources via D-SPCA and ontology-driven reasoning were confirmed on large-scale simulations (100500 IoT devices) across cloud, edge and device layers.

4.2 Performance analysis

Table 1 compares AgroSAGE ontology with state-of-the-art models, including the semantic scope covered, how well the framework works with different protocols, and how suitable it is for edge deployment. Table 2 shows how AgroSAGE and other methods handle the sensor data. It compares the number of sensors used, the accuracy, and the delay reduction.

Table 1. Agriculture-specific ontologies – semantic coverage & deployment suitability.

Ontology	Semantic Scope	Protocol Support	Edge Deploy ability	Annotation Strategy
AgroVoc	Crop taxonomy, practices	None	Low	Static RDF triples
SSN/SOSA	Sensor observations	Partial (HTTP)	Medium	Lightweight RDF
SAREF4Agri	ETSI-aligned agri devices	MQTT	Medium	OWL-based mappings
QUDT	Quantities & units	None	Low	Unit conversion RDF
AgroSAGE Ontology	Crop, Sensor, Phenology, Action	MQTT, CoAP, HTTP	High	Rule-based + GNN embeddings

Table 2. IoT Frameworks & dimensionality reduction – latency, sparsity, and accuracy.

Method/Framework	Dimensionality Strategy	Sensors Used	Explained Variance	MAE (CD1/CD2)	Latency Reduction	Deployment Type
SPCA	Fixed sparse PCA	33	37.2%	0.889/0.689	40%	Cloud
PP-SPCA	Probabilistic sparse PCA	25	49.26%	0.899/0.699	35%	Cloud
ILP + centralized RDF reasoning	GNN + centralized RDF	28	42.1%	0.901/0.712	30%	Cloud
Fuzzy logic + WSN-based zoning	SHACL + SPARQL rules	30	45.3%	0.895/0.705	32%	Cloud
AgroSAGE (D-SPCA)	Dynamic Sparse PCA + GNN	22	49.62%	0.884/0.683	66%	Edge + Cloud

4.3 Ablation study

The Figure 2 shows that the individual contributions of different components in AgroSAGE affect performance. This helps explain the contribution of each module to speed and interoperability.

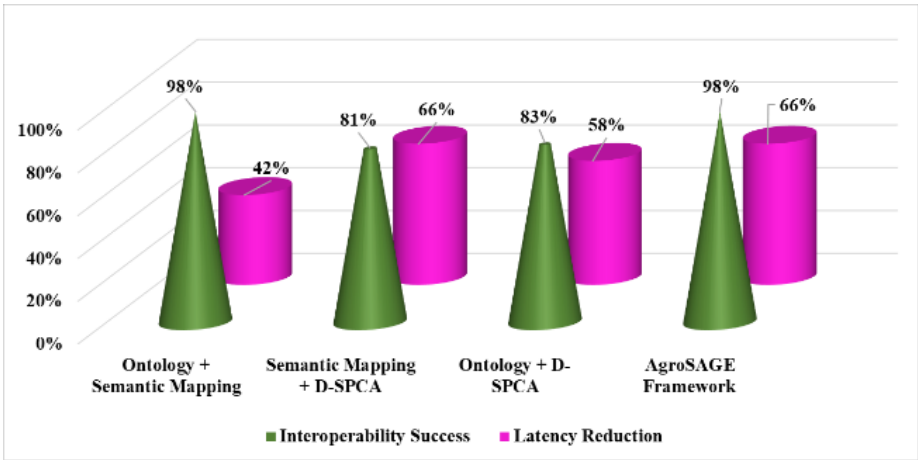

Fig. 2. Graphical representation of ablation study on AgroSAGE framework.

4.4 Comparative analysis

Table 3 compares the existing models of smart agriculture based on their respective Artificial Intelligence (AI) models, deployment context, optimization goals, semantic layer, and Evaluation & Reproducibility of the proposed AgroSAGE framework.

Table 3. Comparison analysis of existing models against the proposed AgroSAGE framework.

System	AI Models	Deployment Context	Optimization Goals	Semantic Layer	Evaluation & Reproducibility
McKinsey 7S-inspired Edge-Cloud Framework [11]	Rule-based (GDD, VPD)	Climate-resilient, low-cost field deployment	Irrigation estimation	None	Moderate; reproducible but lacks semantic depth
Microservices-based Fertigation	ILP-based planning agent	Multi-species fertigation in greenhouse	Nutrient scheduling	Crop-specific logic only	High; greenhouse trials, modular orchestration

Optimization System [12]					
AoI-Aware Energy-Efficient SFC in UAV-Aided Smart Agriculture [13]	FL-DQN + DEC-POMDP	UAV relay, AoI-aware sensing	AoI minimization, energy efficiency	Symmetry-aware RL (non-ontological)	High; formalism-driven reproducibility
IoT-Driven Real-Time Soil and Crop Optimization System [14]	App-based heuristics	Rice fields, Pakistan	Fertilizer recommendation, disease detection	None	Low; app logic, limited editorial clarity
IoT-Enabled Hybrid ML Framework for Smart Agriculture [15]	Ensemble classifiers	Dry beans, soil classification	Crop classification, anomaly detection	None	High; benchmarked accuracy (up to 97%)
AgroSAGE framework (Proposed)	Rule-based + GNN + NLP	Greenhouse + open-field; heterogeneous sensors	Semantic scheduling, energy-aware sensing	SHACL + RDF + SPARQL	High; modular SHACL rules, reproducible annotation

4.5 Discussion

The simulation findings reveal that the proposed semantic- and ontology-based framework effectively contributes to improved interoperability, latency, and scalability in heterogeneous IoT settings. The success of a 98% interoperability level across the MQTT, CoAP, and HTTP protocols confirmed the functionality of the semantic mapping layer. The 66 % %reduction in latency after processing on the edge indicates the importance of local computation in real-time applications. In addition, the system had 85 percent efficiency even with 500 devices, demonstrating robust scaling. These results indicate that structured ontologies combined with distributed computation can overcome long-standing issues in IoT automation. However, such reliance on pre-defined ontologies can restrict flexibility in dynamic or multidomain implementations. Moreover, although semantic translation can enhance communication between devices, its effectiveness in high-data-rate streams or within limited hardware has yet to be determined. Collectively, the outcomes demonstrate the strength of the framework; however, future versions should focus on changing ontology evolution and deployment restrictions.

5 Conclusion

This research proposes AgroSAGE, a modular, ontology-grounded framework that enhances IoT system interoperability and automation. The combination of semantic mapping, protocol reconciliation, and edge-cloud processing allows the system to provide seamless data movement between various devices and enables real-time decision making. The simulation confirmed the critical performance indicators: 98% interoperability success rate, 66% improvement, and maintained efficiencies with 500 devices. These findings highlight the scalability of the framework, retaining the responsiveness and balance of resources. In contrast to baseline systems, the proposed approach provides more rapid communication, decision-making accuracy, and adaptability to various protocols. Processing with structured ontologies guarantees consistent interpretation of each data, and processing edges saw a reduction in delay and de-clouding dependency. This is a significant improvement in the IoT architecture, especially in the areas of smart agriculture, healthcare, and industrial automation. Future developments will aim to incorporate AI-powered dynamic ontologies to enhance adaptability, implement the framework in live environments, and make it more secure when dealing with sensitive information. These steps enable the transformation of the system into a truly autonomous and context-aware IoT element.

References

1. A. Morchid, B. Et-taibi, Z. Oughannou, R. El Alami, H. Qjidaa, M.O. Jamil, E.M. Boufounas, M.R. Abid, IoT-enabled smart agriculture for improving water management: A smart irrigation control using embedded systems and Server-Sent Events. Sci. Afr. **27**, e02527 (2025). https://doi.org/10.1016/j.sciaf.2025.e02527
2. S. Mandal, A. Yadav, F.A. Panme, K.M. Devi, S.K. Shravan Kumar, Adaption of smart applications in agriculture to enhance production. Smart Agric. Technol. **7**, 100431 (2024). https://doi.org/10.1016/j.smartag.2024.100431
3. H.R. Hasan, A. Musamih, K. Salah, R. Jayaraman, M. Omar, J. Arshad, D. Boscovic, Smart agriculture assurance: IoT and blockchain for trusted sustainable produce. Comput. Electron. Agric. **224**, 109184 (2024). https://doi.org/10.1016/j.compag.2024.109184
4. B. Et-Taibi, M.R. Abid, E.M. Boufounas, A. Morchid, S. Bourhnane, T.A. Hamed, D. Benhaddou, Enhancing water management in smart agriculture: A cloud and IoT-based smart irrigation system. Results Eng. **22**, 102283 (2024). https://doi.org/10.1016/j.rineng.2024.102283
5. Y.T. Ting, K.Y. Chan, Optimising performances of LoRa based IoT enabled wireless sensor network for smart agriculture. J. Agric. Food Res. **16**, 101093 (2024). https://doi.org/10.1016/j.jafr.2024.101093

6. S. Garg, N.P. Rumjit, S. Roy, Smart agriculture and nanotechnology: Technology, challenges, and new perspective. Adv. Agrochem. **3**, 115-125 (2024). https://doi.org/10.1016/j.aac.2023.11.001
7. T. Ara, J. Ambareen, S. Venkatesan, M. Geetha, A. Bhuvanesh, An energy efficient selection of cluster head and disease prediction in IoT based smart agriculture using a hybrid artificial neural network model. Meas.: Sens. **32**, 101074 (2024). https://doi.org/10.1016/j.measen.2024.101074
8. A.L. Duguma, X. Bai, How the internet of things technology improves agricultural efficiency. Artif. Intell. Rev. **58**, 63 (2024). https://doi.org/10.1007/s10462-024-11046-0
9. F. Irwanto, U. Hasan, E.S. Lays, N.J. De La Croix, D. Mukanyiligira, L. Sibomana, T. Ahmad, IoT and fuzzy logic integration for improved substrate environment management in mushroom cultivation. Smart Agric. Technol. **7**, 100427 (2024). https://doi.org/10.1016/j.atech.2024.100427
10. X. Li, B. Hou, H. Tang, B.A. Talpur, Z. Zeeshan, U.A. Bhatti, J. Liao, J. Liu, B. Alabdullah, I.S. Al Naimi, Abnormal crops image data acquisition strategy by exploiting edge intelligence and dynamic-static synergy in smart agriculture. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. **17**, 12538-12553 (2024). https://doi.org/10.1109/JSTARS.2024.3414306
11. M. Nawaz, M.I.K. Babar, IoT and AI for smart agriculture in resource-constrained environments: challenges, opportunities and solutions. Discover IoT **5**, 24 (2025). https://doi.org/10.1007/s43926-025-00119-3
12. T. Adamo, D. Caivano, L. Colizzi, G. Dimauro, E. Guerriero, Optimization of irrigation and fertigation in smart agriculture: An IoT-based micro-services framework. Smart Agric. Technol. **11**, 100885 (2025). https://doi.org/10.1016/j.atech.2025.100885
13. M. Akbari, A. Syed, W.S. Kennedy, M. Erol-Kantarci, AoI-aware energy-efficient SFC in UAV-aided smart agriculture using asynchronous federated learning. IEEE Open J. Commun. Soc. **5**, 1222-1242 (2024). https://doi.org/10.1109/OJCOMS.2024.3363132
14. H. Shahab, M. Naeem, M. Iqbal, M. Aqeel, S.S. Ullah, IoT-driven smart agricultural technology for real-time soil and crop optimization. Smart Agric. Technol. **10**, 100847 (2025). https://doi.org/10.1016/j.atech.2025.100847
15. M. Aldossary, H.A. Alharbi, C.A.U. Hassan, Internet of Things (IoT)-enabled machine learning models for efficient monitoring of smart agriculture. IEEE Access **12**, 75718-75734 (2024). https://doi.org/10.1109/ACCESS.2024.3404651