



ARTICLE

Multi-Agent Reinforcement Learning Based Context-Aware Heterogeneous Decision Support System

Taimoor Hassan¹, Ibrar Hussain^{1,*}, Hafiz Mahfooz Ul Haque², Hamid Turab Mirza³,
Muhammad Nadeem Ali⁴, Byung-Seo Kim^{4,*} and Changheun Oh⁴

¹Department of Software Engineering, Faculty of Information Technology, University of Lahore, Lahore, Pakistan

²Department of Software Engineering, Faculty of IT & CS, University of Central Punjab, Lahore, Pakistan

³Department of Computer Science, Faculty of Information Science & Technology, COMSATS University Islamabad, Lahore Campus, Lahore, Pakistan

⁴Department of Software and Communications Engineering, Sejong Campus, Hongik University, Sejong-City, Republic of Korea

*Corresponding Authors: Ibrar Hussain. Email: ibrar.hussain@cs.uol.edu.pk; Byung-Seo Kim. Email: jsnbs@hongik.ac.kr

Received: 10 December 2025; Accepted: 17 February 2026; Published: 09 April 2026

ABSTRACT: The expeditious proliferation of the smart computing paradigm has a remarkable upsurge towards Artificial Intelligence (AI) assistive reasoning with the incorporation of context-awareness. Context-awareness plays a significant role in fulfilling users' needs whenever and wherever needed. Context-aware systems acquire contextual information from sensors/embedded sensors using smart gadgets and/or systems, perform reasoning using reinforcement learning (RL) or other reasoning techniques, and then adapt behavior. The core intention of using an RL-based reasoning strategy is to train agents to take the right actions at the right time and in the right place. Generally, agents are rewarded for the correct actions and punished for incorrect actions. In an RL deployment setting, agents intend to get cumulative maximal rewards through the continuous learning process. These systems often operate in a highly decentralized environment and exhibit complex adaptive behavior. However, the agent's actions on the imperfect nature of context may cause inconsistent reasoning behavior in terms of the agent's reward policies. In this paper, we present a semantic knowledge-based Multi-agent Reinforcement Learning (MARL) formalism for a context-aware heterogeneous decision support system. This is a four-layered architecture to schedule user's routine tasks where user's data is acquired with limited or no human intervention and perform operations autonomously based on agent's reward/punishment policies. For this, we develop a comprehensive case study considering three different domains' ontologies; namely, Smart Home, Smart Shopping, and Smart Fridge Systems, with the prototypal implementation of the system and show the valid execution dynamics, correctness behavior, and verify the agent's optimal reward policies.

KEYWORDS: Multi-agent reinforcement learning; Markov decision process; heterogeneous systems; ontologies; reinforcement learning; context-awareness

1 Introduction

The drastic evolution of Artificial Intelligence (AI) driven e-commerce and the increasing complexity of consumer preferences have opened many facets of intelligent shopping systems to streamline consumer purchasing processes and enhance user experiences. With the advent of AI-driven technologies, these systems provide real-time monitoring and control for better prediction of future needs of consumers and supply chain management systems. Recent years have witnessed the rapidly changing consumer preferences from a traditional to a smart devices dependent life style. Consumers and users often prefer their tasks to be completed autonomously on their behalf, with little or no human involvement. The rise of smart shopping

systems has ushered in numerous opportunities in expanding the realm of grocery management with the assisted systems such as smart grocery buckets or smart refrigerators equipped with Internet of Things (IoT) capabilities, which provide real-time inventory management and enable users to track the product status and receive recommendations for new purchases in case of need. The literature has revealed different approaches of smart shopping systems with the incorporation of conventional decision support systems using e-Commerce, AI-driven autonomous systems, machine learning based predictive systems, etc. [1–4]. Among others, reinforcement learning has emerged as a promising approach that has the capability to learn and adapt user preferences, optimize consumer recommendations, and improve overall consumer satisfaction rate for a personalized shopping environment due to its capability of production based on reward policies.

Reinforcement learning (RL) has been considered as one of the most promising approaches in machine learning, particularly for complex sequential decision making. In the reinforcement learning paradigm, intelligent agents interact with their environments to optimize actions and maximize cumulative rewards [5]. In such systems, agents are usually trained and develop control policies to take optimal actions independently and/or collaboratively. Literature has witnessed numerous studies employing reinforcement learning-based strategies to develop formalisms and applications in different domains, including smart homes and smart shopping systems as independent knowledge base systems highlighting their impact on modern living and consumer behavior [4,6,7]. In [4], Vadivel et al. have proposed shopping experience with the incorporation of reinforcement learning and Q-Network methods to improve navigation efficiency, exploitation trade-off and convergence rate. The proposed smart shopping cart with IoT-enabled framework uses a Reinforced Q-Network with Long Short-Term Memory (ReQL-Net) to optimize in-store navigation by utilizing the temporal context to facilitate adaptive and personalized shopping experiences and enhance the efficiency and convergence performance in a dynamic retail environment. Authors, in [6], have presented a data-driven home energy management (HEM) framework for smart homes by integrating multiple energy resources. They have incorporated a safe reinforcement learning approach using a primal-Dual Optimization based policy search to minimize energy costs to ensure zero constraint violations. Moreover, a combination of RL and Model Predictive Control has been considered to coordinate the scheduling of domestic refrigerators to effectively respond to demand and minimize peak consumption and preserve operational and food safety limits, and provide significant energy savings and grid stability improvement [7]. However, our approach is novel in the sense that the proposed contribution focuses on RL based decision-making in context-aware heterogeneous environments. It has gained very little attention, specifically when it comes to personalized recommendation systems. In this paper, we present a state-of-the-art approach for multi-agent reinforcement learning (MARL) based context-aware intelligent decision support formalism to fulfill user personalized needs autonomously whether the user is at home or in the shopping mall for groceries. This is a four-layered architecture in which a user initially schedules planned tasks for routine activities on their gadget. The first layer is the user interface layer, where user data is collected with partial or no human intervention using smart gadgets. The second layer is the MARL layer, which acts as a bridging layer between Graphical User Interface (GUI) layer and context modeling layer. This is the core layer where reinforcement learning agents perform operations autonomously based on the agent's optimal reward/punishment policies. The third layer is the context modeling layer, in which three independent domain ontologies are developed, namely, smart home, smart shopping, and smart refrigerator system. The fourth layer is the environment layer, where sensors and actuators can be installed through which raw facts are acquired for the context management process. Considering a comprehensive case study of three different domain ontologies, we develop a prototypal implementation of the framework using the Flask and Tensorflow library for model training and evaluate the system's performance metrics by carefully analyzing system's behavior in terms of agents' optimal reward policies.

The remainder of the paper is structured as follows. In [Section 2](#), we briefly review the relevant literature. [Section 3](#) presents the semantic knowledge formulation to contextualize heterogeneous knowledge

sources using a case study of three independent but interconnected ontologies. In [Section 4](#), we present the core concepts of RL paradigm and Markov Decision Process (MDP) along with propose the state-of-the-art MARL approach for context-aware intelligent decision support systems. [Section 5](#) shows the implementation details. [Section 6](#) discusses the significance of the reported work and concludes in [Section 7](#).

2 Related Work

The literature has revealed a renewed research interest in investigating different approaches to the development of smart systems using the notion of context-aware with the incorporation of RL techniques. With this, researchers have emphasized how adaptive learning can better meet the needs of users, resulting in more attention to ease the lifestyle. In [\[8\]](#), a prototype of a context-aware smart home is provided in the simulation, which optimizes by intelligently managing the lighting and televisions according to the environmental conditions. The system helps to preserve energy by considering location-specific data such as the amount of natural light, the occupancy in the bedrooms, and the general presence of the residents within the house. The literature has witnessed that smart shopping has significantly impacted personalized shopping experiences, which allow us to shop more efficiently in less time while achieving greater user satisfaction. In [\[9\]](#), the study explores the design, implementation, and user evaluation of two types of intelligent learning agents for smart shopping carts: utility-based and goal-oriented agents. These agents are developed to improve the shopping experience by aligning with user preferences, tracking shopping behavior, managing personalized lists, and providing real-time guidance and promotional alerts. The system can be integrated into individual shopping carts or advanced holographic displays, offering enhanced user interaction. It uses sophisticated algorithms, including k-d decision trees and reinforcement learning techniques, enabling agents to make intelligent decisions in a partially observable, cooperative, and deterministic environment that incorporates random elements. The framework also incorporates supporting technologies such as text-to-speech, speech recognition, Bluetooth Low Energy (BLE) connectivity, and a picture exchange communication system to cater to users with diverse needs. Machine learning methods empower users of varying abilities to navigate retail environments more effectively. Preliminary user feedback sheds light on their experiences and preferences. The study [\[9\]](#) further examines numerous embodiments of shopping agents, such as holographic projections and social robots, highlighting the practicality and convenience of a robotic shopping cart capable of multiple users. Special attention is paid to the performance of the Q-learning algorithm in real-world applications, analyzing the factors that influence its learning process and ongoing efforts to enhance agent development.

The shopping cart is a very essential part of a smart shopping system and offers several features, including offering product suggestions, tracking spending cost, and suggesting the most optimal route. The study [\[10\]](#) provides a clear exposition of the Intelligent Smart Shopping Cart Learning Agents prototype with its operating environment. The use of machine learning will be combined with beacon-based and hologram-assisted navigation to enhance the shopping center experience of customers. A Picture Exchange Communication System (PECS) increases accessibility by allowing interaction via drag-and-drop interactions and keyboard shortcuts. The system operates in a partly observable, cooperative, and multi-agent system, and is mostly deterministic but with some stochastic component to retain dynamism and adjustability. The episodic task space can take a semi-dynamic or a static form depending on situational circumstances. It is important to note that the agent is based on a k-d decision tree to enable effective monitoring of shopper places as well as delivery of relevant promotions based on the shopper list. There is another agent that uses reinforcement learning to minimize routing and maximize overall efficiency. Together, a personal shopping utility agent learns user-specific behaviors in order to narrow down to individualized strategic plans. Some of the most important performance indicators include precise customer

navigation, increased user comfort, and navigation assistance that is significantly beneficial to people with communication difficulties in large retail stores. A smaller sample size-based empirical study produced a positive reaction, with respondents providing a positive account of the efficiency of both the personal utility agent and the promotion-based learning agent in enhancing the shopping experience.

In addition, the identified research gap is related to the lack of a unified semantic knowledge driven MARL framework that can effectively facilitate context-aware, adaptive, and autonomous decision-making in heterogeneous smart environments. Above mentioned existing approaches lack consistent reasoning and seamless interoperability among diverse domains, particularly under uncertain and dynamic contextual conditions. Therefore, there is a crucial need for a comprehensive framework that integrates semantic knowledge representation with RL to achieve scalable coordination, contextual adaptability, and optimal policy learning in decentralized multi-agent systems. The complete proposed framework explanation is provided in further [Sections 3–6](#).

3 Contextualizing Distributed Knowledge Bases for MARL Formalisms

The development of a distributed knowledge base has gained significant attention in recent years due to Distributed Description Logics (DDL) and their semantic inter-ontology mapping [11–14]. Interontology mapping privileges different ontology axioms to interconnect using Distributed Description logics. DDL is a formal logical framework for conceptualizing different domains in description logic (DL) knowledge bases to model and reason heterogeneous information. In DDL, different DL knowledge bases are interlinked to exchange information. As DL axioms can be semantically mapped to the corresponding ontology axioms, semantic mapping of DL knowledge bases is trivial by interlinking ontologies [15]. The core notion of using DDL is that it preserves the identity, uniqueness, and independence of each DL knowledge base structure while exchanging information among different ontologies. Each ontology knowledge base consists of two kinds of axiom known as Terminology Box (TBOX) and Assertion Box (ABOX). TBOX represents the terminology of the domain, which is also expressed as entities and relations in the ontology. The ABox represents the assertions about individuals or instances of the classes along with their relationships. Different domain ontology axioms can be correlated to form interontology axioms, also known as bridge rules, which are often expressed as: $C_a \sqsubseteq D_b$; where C_a, D_b are concepts (TBOX) of ontologies O_a and O_b , respectively. It is often expressed in a tuple as $\langle \mathcal{T}, \mathcal{A} \rangle$ where \mathcal{T} represents the set of TBoxes and \mathcal{A} represents the set of distributed ABoxes. If we consider a set of DLs knowledge bases, then every DL knowledge base is specified by $\{\mathcal{DL}_i\}$, where $i \in I$ is a part of a non-empty set to indicate individualized ontologies.

In the literature, several knowledge base systems and knowledge representation techniques have been proposed for modeling and reasoning human-like assisted decision making and inferencing systems. The ontology-driven knowledge-base systems have been considered to be the most optimistic approach due to its simplistic, flexible, expressive, and efficient reasoning capabilities [16,17]. According to Gruber [18], ontology is formally defined as “an ontology is an explicit specification of a conceptualization”. Ontology can be used to explicitly model the domain using its language constructs such as entities (classes) along with their relationships. The Literature has witnessed two different variants, namely; OWL 1 (Web Ontology Language) and OWL 2 [19]. Both OWL 1 and OWL 2 have sub languages for custom-built solutions according to their reasoning capabilities, efficacy, expressivity, and scalability of the system. OWL 2 Rule Language (OWL 2 RL), a sub language of OWL 2, is specifically designed to express rule-based axioms of ontologies. The Semantic Web Rule Language (SWRL) is a custom built feature to express more complex rules for real-time problem solving. SWRL is basically founded on the OWL rule languages that determines the ability to write Horn like rules in OWL concepts and roles. It can make good arguments with the OWL. This can be achieved with SWRL, however, as OWL languages do not always have the capabilities of expressing all sorts of relations. OWL 2 RL rules are written in ontology using the expressions of class and sub-class

relationship and property and sub-property relationship. OWL 2 RL combined with SWRL has a greater expressive power than OWL 2. SWRL rules allows users to write customized user-defined rules which enhances expressivity and completeness of logical axioms for sound reasoning capability. The ontology axioms expressed in the form of OWL 2 RL and SWRL rules format, which can be transformed into a plain text in the horn-clause format using approaches [12,14]. The transformed format is interpreted in the first order axiom pattern and is the most suitable representation for designing and developing rule-based multi-agent systems. As a rule-based multi-agent system collaboratively performs reasoning using a set of rules derived from ontologies to perform dedicated tasks as per custom designed rule-based system. A rule-based multi-agent system consists of a knowledge base and a working memory. A knowledge base is a set of rules, and working memory is a set of facts used by those rules in their instantiations. For example, $Person(?p), AmountDeductFromShoppingCard(?p, ?sc) \rightarrow hasAlertonMobile(?p, "Amount Deducted with Details")$: This rule specifies that when a person (?p) is identified and an amount is deducted from the associated shopping card (?sc), a mobile alert is generated to inform the user of the transaction details. In this work, we develop a comprehensive case study by constructing three different domains ontologies, namely; Smart Home, Smart Fridge, and Smart Shopping system and their OntoGraph are presented in Figs. 1 and 2. This case study is specifically designed to help users in their convenient zone while staying at home and/or shopping in the market. At this stage, we are intending to provide the design of the system with domotic features. For example, the design of a smart home system can be equipped with different sensors, such as a lighting and temperature control system and embedded devices to detect the presence of inhabitants and turn on/off the appliances according to the custom need or presence of the users. The smart home operates according to predefined rules. The smart fridge system will autonomously assist users in identifying which goods are short or expired and need to be purchased using an autonomous response generator. The smart shopping system autonomously assists users in developing the grocery list using the smart fridge while the user is visiting the market or when needed. Users can be connected to the smart trolley system in the supermarket using their customer cards (Rules shown in Fig. 3). As the product is added to the trolley, its price will accumulate in the bill. When customers approach the smart billing counter, instead of waiting in a long queue, they can directly pass the smart billing passage, which can automatically save time in billing.

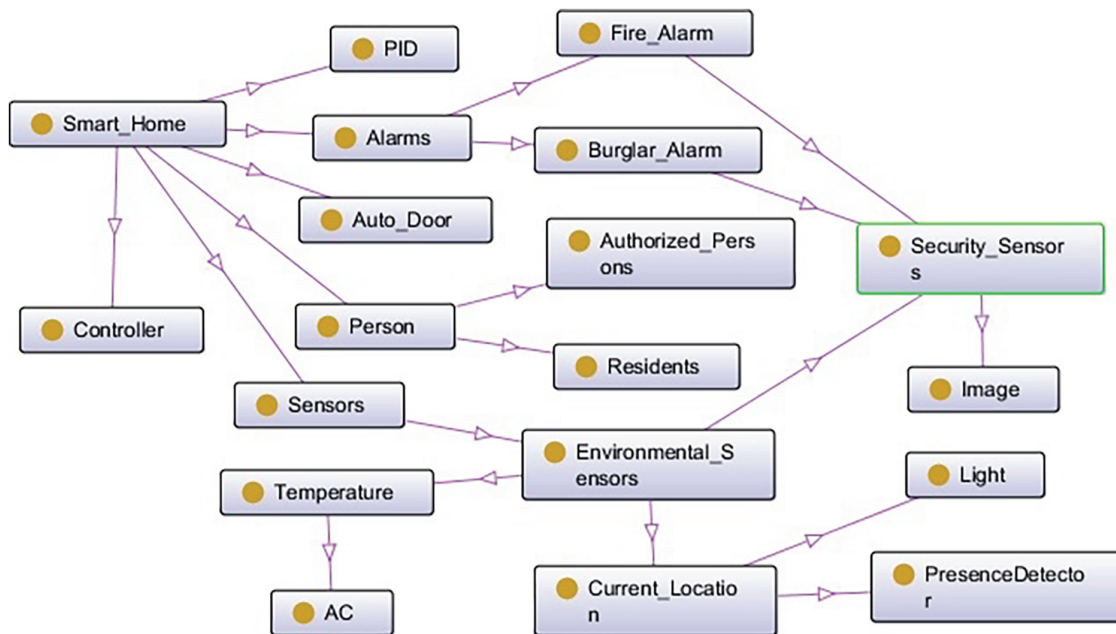


Figure 1: OntoGraph of Smart home system.

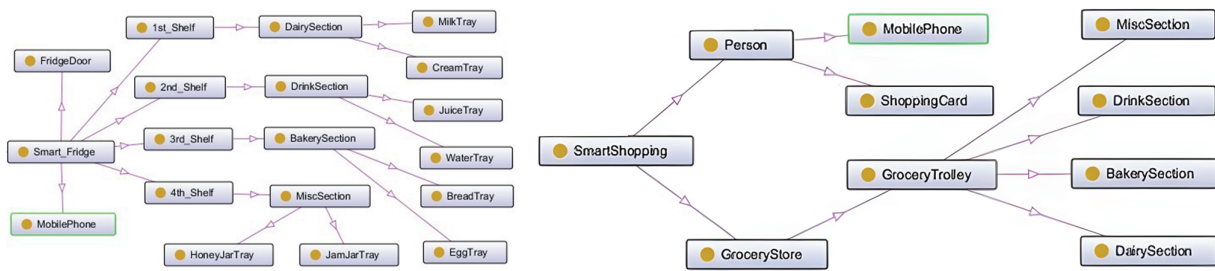


Figure 2: OntoGraph of Smart fridge and shopping system.

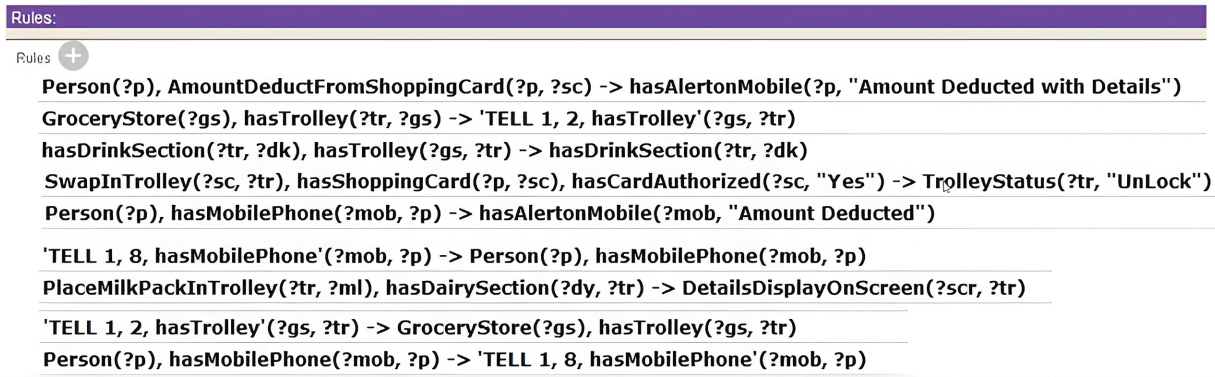


Figure 3: Rules of Smart shopping system.

4 MARL Based Context-Aware Reasoning Formalism

In this section, we present a MARL based context-aware reasoning formalism for heterogeneous decision support systems. In this formalism, we show how context-awareness and understanding of dynamic environments involve analyzing and interpreting sensor data such as coordinates, time, weather, and user activities collected via smart/embedded devices. This contextual data is processed through the context modeling layer that incorporates ontologies, semantic reasoning, and lightweight machine learning techniques to generate high-level contextual information. This semantic information is transformed into a plain text of horn-clause rules (agent's understandable format), enabling them to support suitable inferencing and act according to their environment. As a result, agents can make context-aware decisions and dynamically adapt to environmental changes using predefined rules and interaction protocols. Reinforcement learning plays a crucial role in this pipeline by enhancing agent's performance through feedback loops. RL is a subset of machine learning that focuses on training agents to perform optimal actions in a sequential pattern with the intention of obtaining the maximum cumulative rewards. This capability is aligned with the systems where smart shopping related agents learn from customer preferences or from environments and apply best feasible policies with a customized settings to gain ultimate customer satisfaction. One of the key benefits of using RL is its quantitative analysis techniques and anticipated decision-making capability in which learning agents provide ease to consumers in making optimal decisions during shopping. RL algorithms have dynamic interactive capability to continuously optimize the policy based on self-learning behavior of the agents. The main structure of reinforcement learning is described in Fig. 4 [20]. The core intention of using RL agent is to find optimal policy for attaining maximal cumulative rewards [21]. The agents in the system are guided by the RL approach, so every action made by any agent receives either a reward or punishment. At a successful action, the agent gets reward r by the environment following action a in the state s . The reward

mechanism can be represented as follows where $R(s, a, s')$ represents the reward achieved after leaving state s and moving to state s' . When agents execute idle transitions, then no reward can be received by the system. In Reinforcement Learning, two fundamental equations are commonly used: one is called the State Value Function, and the other is known as the Action Value Function. The state value function (mentioned in Eq. (1)) gives the expected total future reward when the agent starts from state s and follows policy π thereafter. The action value function (mentioned in Eq. (2)) measures the expected total future reward when the agent starts from state s , takes action a , and then follows policy π .

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s \right] \quad (1)$$

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, a_0 = a \right] \quad (2)$$

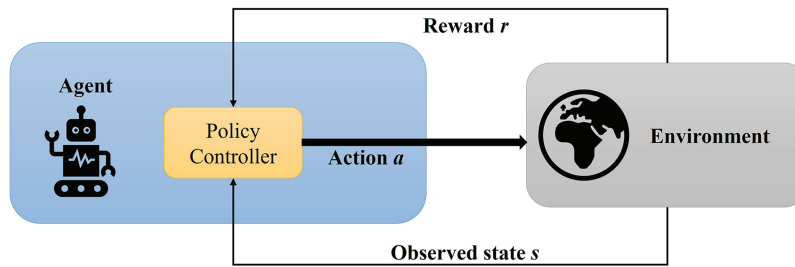


Figure 4: Reinforcement learning.

The key components of the state value function that are shown in Eq. (1) are: s_t state at the time step t . a_t action chosen at time t following the policy $\pi(a_t|s_t)$. $R(s_t, a_t)$ reward after taking action a_t in state s_t . γ discount factor (between 0 and 1). \mathbb{E}_π expects all possible future paths when following the policy π . In Eq. (2), $s_0 = s, a_0 = a$ are the initial conditions that define the starting state and the action from which the return is calculated. $\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$ are the cumulative discounted reward (return) that the total reward accumulated over time, discounted by how far in the future it occurs. In the RL paradigm, agents are trained to make optimal decisions from a set of actions and receive feedback on these actions in the form of reward and punishment. Agents learn to take the best actions based on their experience to maximize rewards and minimize penalties. Agent's executions are performed in the form of states and actions for complex problem-solving and sequential decision making, which is denoted by a mathematical structure known as MDP. In MDPs, decision-making strategies are developed using a specified set of rules known as a policy. The policy provides guidelines on a set of actions that agents take to attain the desired goals. In the literature, MDP is often represented by a tuple with a customized combination; we use the MDP tuple as (S, A, P, r, γ) where S is a set of states, denoted as $S = \{s_1, s_2, \dots, s_n\}$. A is a set of actions performed by agents, denoted by $A = \{a_1, a_2, \dots, a_n\}$. P is a transition function in the form of a Cartesian product ($P : S \times A$) that maps the corresponding action from state s_t to the next state s_{t+1} . R is a set of rewards for each corresponding action against a transition from state s_t to s_{t+1} with possible action a_t . γ is a set of discount factors ranging from 0 to 1 based on the execution behavior of the policies and agents [22]. MDPs offer a robust framework for modeling scenarios in which the environment is uncertain and influenced by both stochastic factors and the actions of decision makers. In MDP, system outcomes are partially random and partially under control based on decision-making scenarios. The illustrations of commonly used MDP components are as follows:

$$M = (S, A, P, R, \gamma)$$

1. States (S): States that include all the possible situations in which the agent could exist. All the data needed to make a decision is collected by each state.
2. Actions (A): The agent has a limited list of possible actions at any time. At every stage, the agent can choose from the actions available to change its environment.
3. Transition Probability (P): A probability function $P(s'|s, a)$ that describes the probability of transitioning from state s to state s' after capturing action a . This compresses the dynamics of the environment.
4. Reward (R): A reward function $R(s, a, s')$ awards a scalar reward against each agent's action from state s to s' . The reward is the accumulated towards expected outcomes.
5. Discount Factor (γ): A discount factor $\gamma \in (0, 1)$ that regulates the position of future rewards. A discount factor close to 1 makes the agent arrange long-term rewards, while a factor close to 0 makes it emphasis on instant rewards.

In RL deployment setting, with the support of adaptive learning, multiple sensor agents contribute constraints that shape the state representation. The contextualization is based on multiple ontologies and agents that are connected through the SWRL and MARL. These agents utilize semantic knowledge rules to make decisions based on specific domain knowledge, such as user preferences or environmental conditions. Thus, building a contextual view of the environment, such agents act in a multi-context system where they were sharing the information. Within ontologies, SWRL enhances semantic reasoning by encoding complex relationships, while MARL enables agents to learn collaboratively, optimizing their actions in uncertain environments. Meanwhile, this approach makes it possible for agents to use data and collaborate on different systems in dynamic and complex environments. The proposed approach uses MARL to support distributed autonomy, enabling different agents, such as those who control lighting, the fridge, or shopping, to learn optimal policies independently, but maintain coordinated behavior. MARL's ability to adaptively respond to the inherently dynamic and uncertain nature of contextual environments, including fluctuating user preferences, ensures robust system performance. Moreover, the paradigm agrees with the scalability and interoperability of heterogeneous domains in which centralized control is not feasible. In a reward-based optimization model, agents have the freedom to coordinate between multiple agents effectively to achieve a goal.

Smart environments are designed to have three main layers, Perception, Processing, and Action/Control. In the perception layer, an array of sensors and smart devices are used to collect and combine data. For example, the system uses motion detectors and cameras to track the people who are in the building and its energy consumption. The next one is the processing layer that makes the usage of cloud and edge computing with the help of machine learning to use multiple algorithms based on data so that decisions can be made and how users will act can be predicted; with the passage of time, the system should be able to change and improve. At the action/control layer, the decisions are implemented as actual actions using actuators for thermostats, and lighting allows the system to respond automatically in real-time feedback. All components are interconnected, and adaptively the system can work smoothly and handle changes. They work in various smart environments to make it more useful and easier to use and continue to improve their performance.

The framework outlined in Fig. 5 includes multiple layers, and each layer communicates with the other. The proposed architecture is designed in such a way that it has four different but interrelated layers, each playing a vital role in the realization of context-aware decision-making. The GUI Layer, a first layer, provides user data acquisition with little or no human intervention using smart devices and sensors. The second layer is the MARL layer, which interprets the interface between the GUI and the context modeling layers. This is the operational layer in which RL agents act independently to execute tasks under optimized rewards and punishment policies to achieve the goal of intelligent adaptive behavior. The third layer, the

Context Modeling Layer, refers to three domain-specific ontologies smart home, smart shopping, and smart fridge systems. This layer aims to achieve and sustain system heterogeneity with an agreed information exchange mechanism and to guaranty the identity, privacy, and independence of individual domains. The ontologies have been built in OWL2 RL with the Protégé editor, including a full set of facts and rules that allow semantic reasoning. Moreover, the context modeling layer produces domain-specific ontologies, which produce structured facts and inferred knowledge (via entities, relationships, and rules), and transfers them to the MARL layer as knowledge bases (KBs). These derived contextual states are input to MARL agents that inform the representation of states and restrict valid actions and the agents themselves learn optimal policies through reward interactions. Therefore, semantic reasoning can help offer contextual foundations, and MARL can be used to achieve adaptive decision-making and optimize policies. Lastly, the Environment Layer serves as a way of interacting with the physical world in real-time, deployment of sensors and actuators being used to gather raw contextual information to be fed into the context management and reasoning processes. Furthermore, in Fig. 5, first layer is a GUI that interacts with the environment layer, putting Excel data into a database so that updates occur immediately; these data are accessible from the GUI application. The second and third layers consist of a context layer that combines multiple ontologies and can relate to or influence each other. This layer involves certain entities, their relationships, and predefined rules that we have defined separately in this paper. Based on facts and rules, the context modeling layer interacts with the MARL layer. In this, we are mapping all the rules to multiple agents on a knowledge base.

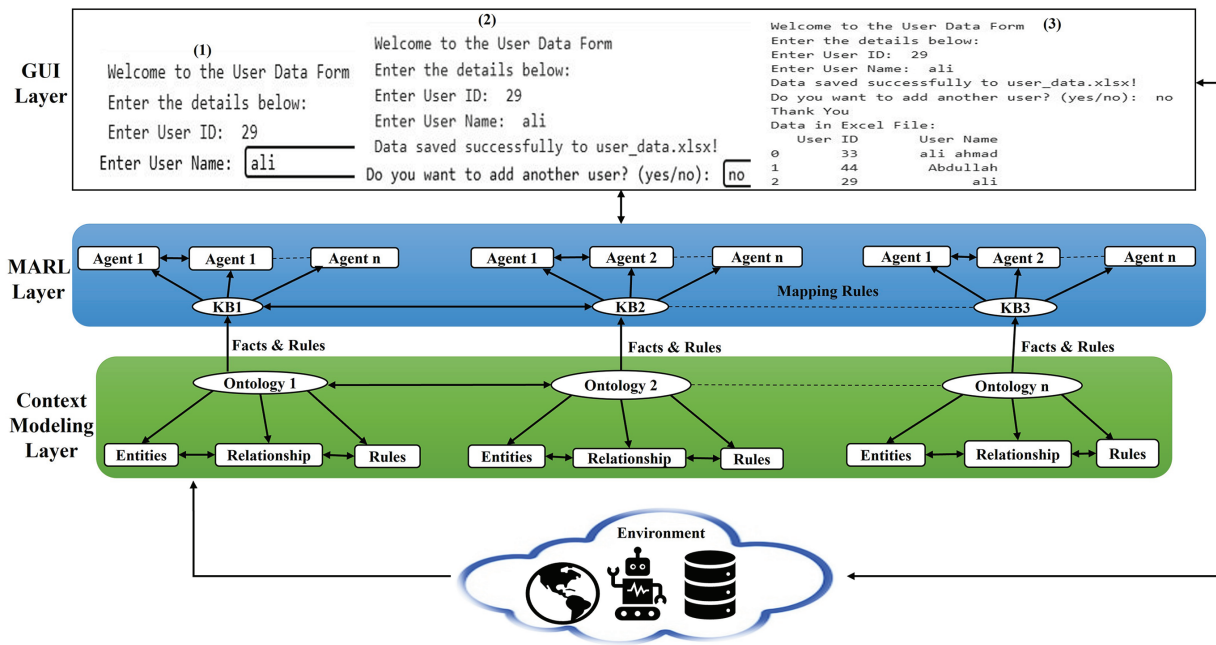


Figure 5: Integrated system framework.

Moreover, in Fig. 5, we interact with the Agent environment, which has multiple agents in place that connect to a database. We are using a front-end/user interface in the system, which we designed in Python, and it interacts with the database to store data. In this way, we are saving data into the database while maintaining the interaction with the system. The GUI layer interacts with the MARL layer, which contains multiple knowledge-based sources (KB1, KB2, KB3). Each knowledge base is equipped with multiple agents operating based on specific predefined rules. These rules have already been defined, and the knowledge bases interact with each other, as shown in Fig. 5. The MARL layer interacts further with the context modeling layer,

where we can see that there are multiple ontologies (from ontology 1 to ontology N). Each ontology contains certain entities, their relationships, and their associated rules that link back to the MARL layer. There is an environment that interacts with both the context modeling layer and directly with the system (front-end) for data processing. In a smart environment, the extraction of information for agent sensor data to support agents is structured, including data acquisition, processing, and context integration to make timely and accurate decisions. The operations of an intelligent system are realized using a variety of smart sensors, including motion detectors, temperature sensors, and smart cameras (i.e., Smart Home System) that continuously capture environmental data. This raw data is pre-processed through noise reduction, data fusion, and feature extraction either at the edge or cloud level to improve its quality and usability. In this paper, the proposed approach further maps the data to an ontology-based context model by structuring sensor readings in predefined knowledge representations and facilitates semantic understanding. In terms of contextual information, it allows multi-agent systems to dynamically adjust their behavior and optimize decision-making in changing environments. Additionally, the proposed framework enhances agent's reasoning capabilities by offering a contextualized ontology. This is achieved through structured and hierarchical knowledge representations, including spatial location, user preferences, environmental conditions, and system constraints. Semantic reasoning technologies such as DL and SWRL (Semantic Web Rule Language) enable agents to propose new knowledge and change their behavior in response to changing environmental conditions. Furthermore, as we see in Fig. 5, the integration of multiple ontologies in a multi-context system is able to achieve the seamless exchange of heterogeneous data sources, thereby ensuring interoperability and beneficial decision making in a highly dynamic and complex environment. As a result, we can leverage the structured approach of intelligent agents to make them more adaptive, more precise, and able to cope with diverse real-world scenarios within the smart environment. In Fig. 5, the integration of ontology data into MARL improves the collaboration between structured semantic knowledge and adaptive learning mechanisms in multi-agent systems for better decision making. MARL agents take as input prior knowledge in the form of a predefined rule set (already defined in the Protégé tool), entity relationships, and constraints in the form of ontologies. Additionally, having semantic data in a structured format within the MARL framework enables agents to reason contextually as they learn optimal policies through reinforcement learning techniques. This integration improves clarity, reduces search space, lowers exploration costs, and supports better adaptability in heterogeneous environments. Furthermore, ontology-driven intelligence is used in MARL-based applications for mobile devices (UI designed in Python) to improve tasks such as dynamic resource allocation, energy efficient management, and personalized user experience.

In this case, applications use distributed agents that interact with each layer in real time, constantly learning from feedback and adjusting to changes in context. These agents can incorporate reinforcement learning, enabling them to coordinate decisions effectively and ensure smooth interactions among different components of the system. Since MARL-based applications take into account user behavior patterns, environmental factors, and system constraints, which can utilize cloud edge collaboration to increase scalability and efficiency [23,24]. Therefore, this framework turns out to be critical for numerous real-world applications such as smart home automation, intelligent navigation systems, context-aware recommendation systems, etc., and MARL will thus be a key enabler of next-generation smart environments [25].

4.1 Integrated Case Studies of Intelligent Smart Systems

Case studies in smart environments, such as smart home, fridge, and shopping systems, highlight the transformative impact of integrating IoT and machine learning technologies. Smart homes optimize energy use, improve security, and improve comfort by automating devices such as thermostats and lighting based on real-time data and user preferences. In addition, smart fridge track inventory and monitor expiration

dates, suggest recipes, and order groceries to reduce waste and match dietary habits. Smart shopping systems personalize the retail experience by analyzing consumer behavior to recommend products, offer customized discounts, and optimize delivery. These examples demonstrate how data-driven decision making improves efficiency, promotes sustainability, and increases user satisfaction, reflecting a shift toward intelligent systems that address our needs while benefiting the environment.

4.1.1 Smart Home System

In this section, we discuss a smart home system (SHS) where multiple agents can be installed and all are interconnected with each other. The primary purpose of this system is to automate various tasks within the smart home and significantly improve security with the help of agents. In this system, if any suspicious or unauthorized person enters the house, an alarm will be triggered (shown in Fig. 6a). Furthermore, when any presence is detected, the installed agents will control the lights and air conditioning (AC) by turning them on or off (shown in Fig. 6b). The implementation of a smart home system in Python with multiple conditions in graphical form is represented in Fig. 6.

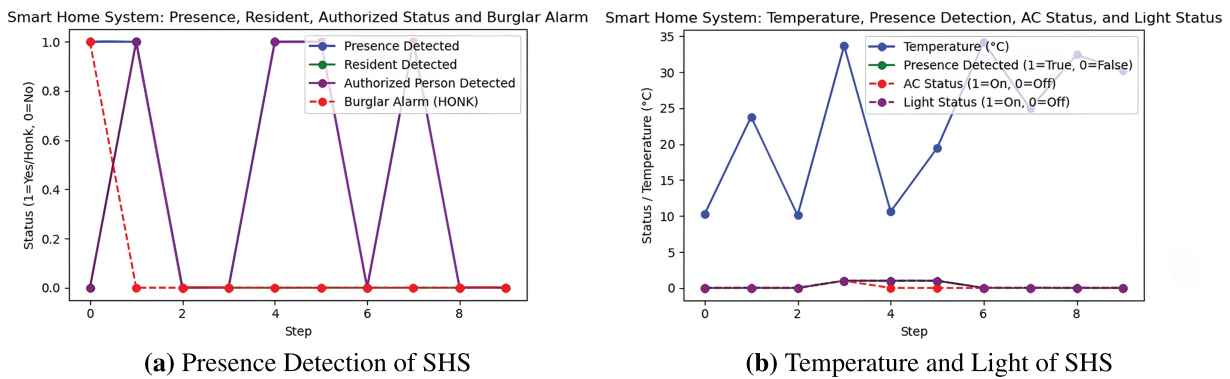


Figure 6: Smart home system.

4.1.2 Smart Shopping System

In this scenario, we discuss a smart shopping system (SSS) that allows users to purchase items suggested by the smart fridge. The system incorporates a smart trolley that provides real-time updates on the quantity and cost of the selected items. The smart trolley is unlocked using a smart shopping card and users can place all necessary items in the trolley. When the user exits the store, the payment is automatically deducted from the shopping card and the user receives an alert that confirms the successful payment message. With the help of Python, the implementation of the smart shopping system is illustrated in Fig. 7a and b.

4.1.3 Smart Fridge System

In the third case study, we discuss a smart fridge system (SFS) designed to alert you in different scenarios. For example, if the fridge door is left open, it will send an alert. Similarly, the fridge has various sections where multiple items, such as milk, jam, honey, eggs, bread, etc., are stored. When any of these items reach a specific minimum level, the system will alert you that the item is about to run out and should be restocked. The implementation of the smart fridge system using Python is illustrated in Fig. 8a.

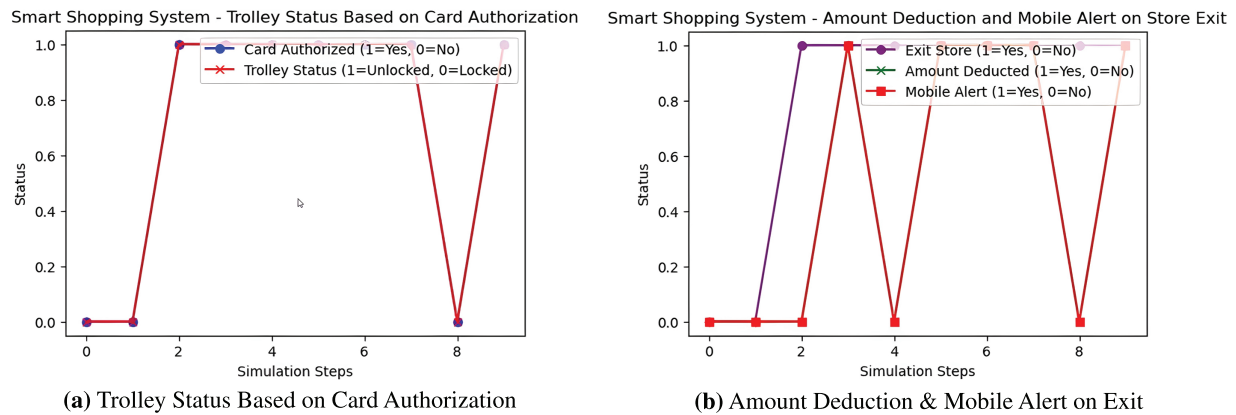


Figure 7: Smart shopping system.

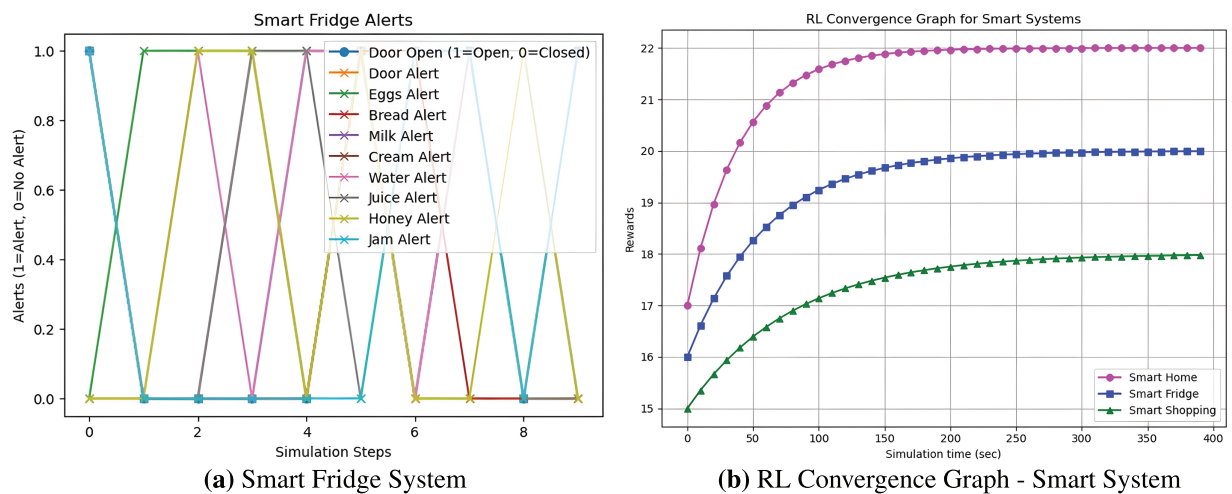


Figure 8: Smart fridge and RL graph.

5 Implementation of MARL Based Context-Aware Framework

In this section, the system is implemented using Python within a Jupyter Notebook environment. The implementation process is summarized in Fig. 5, which outlines the specifications and design steps. Three case studies Smart Home, Smart Fridge, and Smart Shopping System have been evaluated under different environmental conditions. Each case utilizes predefined semantic rules developed using the Protégé ontology editor. These rules are applied to simulate intelligent behaviors, and the response of the system is visualized through graphical representations to demonstrate their effectiveness. The implementation of a smart environment in Python involves the integration of sensor data from the IoT, data processing frameworks, and ML models. Libraries such as Flask and FastAPI are employed for web-based interfaces and high-performance APIs, respectively. For handling and analyzing large datasets, Pandas and NumPy are utilized. RL models, developed using frameworks like Stable-Baselines3 or TensorFlow, enable agents to learn optimal behaviors from sensor data, enhancing real-time decision-making capabilities. The training is episodic, in which the agents play with the environment, monitor the contextual states of semantic reasoning, and revise their policies with regards to reward-punishment feedback. This integrated pipeline supports the development of adaptive smart environments that improve system intelligence and response time. In the

Smart Home scenario (Fig. 6a), the burglar alarm is activated only when an unauthorized presence is detected, as identified by the detection agent. Fig. 6b illustrates that both the AC (specific temperature threshold is met) and the lights are turned on when a resident is detected. In the absence of detected presence, both the AC and the lights remain inactive. The second case study focuses on the Smart Shopping System. As depicted in Fig. 7a, the trolley remains locked unless a valid authentication card is used. Upon successful authentication, the trolley is unlocked, ensuring security and controlled access. Fig. 7b shows that once a customer leaves the store with purchased items, the total bill is automatically deducted and a payment alert is sent to the user's mobile device. This action is conditionally triggered by the exit event. Collectively, these case studies demonstrate the functionality and adaptability of the proposed multi-agent system, which utilizes semantic rules, contextual awareness, and reinforcement learning to enable intelligent decision-making. The ability of the system to operate effectively under dynamic conditions highlights its potential in building robust smart environments. The Smart Fridge System is our third case study, which consists of multiple sections, including:

1. Dairy Section (Milk, Cream)
2. Drink Section (Water, Juice)
3. Bakery Section (Eggs, Bread)
4. Miscellaneous Section (Honey, Jam)

In this system, multiple alerts are generated. Whenever any item in a section falls below a specific quantity threshold, a notification is sent to the user's mobile. In addition, alerts are also received on the mobile based on the status of the fridge's door (open/closed). This behavior is represented in Fig. 8a. Furthermore, the convergence graph shown in Fig. 8b examines the reward stabilization curves in three smart environment agents Smart Home, Smart Fridge and Smart Shopping over simulated time. In this graph, all lines indicate the rewards of the respective agent accumulated up to the steps in which it sequentially improves its decision-making mechanism responding to the feedback it faces in its environment. As shown by the magenta curve with circular markers, Smart Home shows the highest speed and convergence stability, reaching the optimal reward sooner than other agents. Its rapid learning rate shows a successful adjustment to environments such as temperature controls, presence detection, as well as automating the devices, and an efficient security framework. The smart fridge as visualized by the blue curve with square dots converges slower. Its reward profile resembles the dynamical complexity involved in the monitoring of inventories, as well as the various alert states, including the state of the doors and the perishable stock conditions. The system intelligently strikes the balance between providing advance warning and its low false-alert rate, finally balancing out at its theoretical maximum reward. The green curve in which the markers are in the shape of triangles is Smart Shopping and shows the slowest convergence among all three. This pattern is also consistent with the erratic nature of consumer behavior in a business environment, which encompasses multiple variations in the number of approvals of the shopping card, product choice, and purchases. Although this is not constant, the agent stabilizes in the end and shows its ability to provide real-time decision making in dynamic commercial environments.

6 Discussion

This paper explores the implementation of multi-agent systems for automating routine tasks, leveraging MARL integrated with the MDP. In addition, we conducted a comparative analysis of existing approaches with pros and cons presented in Table 1. Although the literature discussed in Section 2 has significantly impacted different directions of thought in smart systems like smart home, smart fridge and smart shopping systems. The proposed system architecture, illustrated in Fig. 5, consists of four layers: user interface, environment layer, and two intermediate layers dedicated to MARL and context modeling, which collectively

facilitate intelligent decision-making and automation. To evaluate the practical utility of the framework, three case studies were selected: Smart Home, Smart Shopping, and Smart Fridge systems reflect real-life applications. These case studies were initially modeled using the Protégé tool, using an ontology-based design to define domain-specific entities and relationships (Figs. 1 and 2). Based on these ontologies, rule-based execution models were developed (Fig. 3) and subsequently implemented in Python using Jupyter Notebook. A user interface was created to allow interaction with the system, with all user data stored in an Excel format. Rule execution was dynamic and system output was visualized through real-time graph updates for each iteration. Overall, the proposed multi-agent system demonstrates adaptability, contextual awareness, and automation capabilities across diverse smart environment scenarios.

Table 1: Comparative analysis of used methodologies.

Paper	Core Methodology	Pros	Cons
[26]	Q-learning, Hierarchical Reinforcement Learning (HRL)	Capturing adaptive behavior, Showing realistic interactions, Use of hierarchical task decomposition	Ignores smart home functionalities, Limited discussion on scalability, Lacks real-world deployment
[27]	Markov Decision Process (MDP)	Cost reduction and comfort maintenance, Model-free design, Continuous action space handling	Real-world deployment challenges, Requires large training data, Single smart home scenario
[28]	Multi-agent reinforcement learning (MARL), Integrated feature extraction & Supervised learning	Reduced data requirements, Significant energy savings, Privacy concerns, Fast convergence	Relied on identical TCL hardware, Requires communication infrastructure
[29]	Multi-agent deep reinforcement learning (DRL), Tunable Deep Q-Network (DQN)	Allows dynamic adaptation, Handles multiple conflicting objectives	Lacks real-world deployment, Limited discussion on communication overhead
[30]	Reinforcement Learning (RL) and a Q-learning algorithm, Action-value functions, Grid-based map modeling	Handles dynamic environmental changes, Enhanced task efficiency and path optimality, Integration of Q-learning	Require extensive episodes and computational time, Communication overhead among multiple agents not thoroughly discussed
[31]	Trust-MARL, Trust modeling mechanism	Combines trust evaluation with MARL, Addresses real-world applicability	Trust modeling is predefined, Trust calculation & joint training has less scale
[32]	Heterogeneous-agent RL, Applied CTDE, Used multi-agent benchmarks	Enhances scalability and realism, Provides a theoretically grounded framework, Supports modularity and flexibility	Requires increased computational overhead, Limited by training complexity and sample efficiency

(Continued)

Table 1 (continued)

Paper	Core Methodology	Pros	Cons
[33]	MARL, Q-learning	Scalable and Decentralized, Minimal Human Intervention, Adaptive to Dynamic Changes	Scalability to Complex Systems, Reward Design Sensitivity

7 Conclusion and Future Work

In this paper, we have proposed a framework that shows how MARL can be included in smart environments. It is shown how this paradigm applies through case studies involving Smart Home, Smart Shopping and Smart Fridge systems. The analysis shows that automating decisions with ontologies and rules allows quick responses and flexibility in changing situations. The system relies on MDP-based learning to automatically adapt its responses, helping to protect the home, make shopping, and manage food. The effectiveness and usefulness of the system are demonstrated by running simulations and visually analyzing the results using Python Flask. The results show that the use of MARL automation noticeably increases the efficiency, security, and user-friendliness of the smart system. In future work, we develop the state-of-the-art application with the intention of handling AI assisted support for autonomous decision-making, and make the system suitable across different platforms.

Acknowledgement: The authors would like to thank all individuals and institutions who contributed to the successful completion of this research. Their support and valuable insights were greatly appreciated. Special thanks to Hongik University new faculty research support fund for supporting this work.

Funding Statement: This work was supported by the Hongik University new faculty research support fund.

Author Contribution: The authors confirm contribution to the paper as follows: Conceptualization, Methodology, Writing—Original Draft Preparation: Taimoor Hassan, Hafiz Mahfooz Ul Haque; Literature Review, Visualization: Muhammad Nadeem Ali, Byung-Seo Kim, Changheun Oh, Hamid Turab Mirza; Reviewing and Editing: Ibrar Hussain. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this study are available from the corresponding authors upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Chabane N, Bouaoune A, Tighilt R, Abdar M, Boc A, Lord E, et al. Intelligent personalized shopping recommendation using clustering and supervised machine learning algorithms. *PLoS One*. 2022;17(12):e0278364. doi:10.1371/journal.pone.0278364.
- Hanif M, Khan M, Acharjee U, Jewel K, Ahamed M. Smart e-commerce shopping: innovations, challenges and future trends. *Int J Bus Soc Sci Res*. 2024;12(1):7988.
- Choppa A, Kesavarshini A, Bhavishya M, Rao YN. Reinventing grocery shopping with reinforcement learning. In: 2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS); 2024 Dec 4–6; Pudukkottai, India. p. 1375–80.

4. Vadivel PS, Karthika B, Robinson YH, Krishnan RS, Rachel L, Sundararajan S. An intelligent IoT-driven smart shopping cart with reinforcement learning for optimized store navigation. In: 2023 International Conference on Emerging Research in Computational Science (ICERCS); 2023 Dec 7–9; Coimbatore, India. p. 1–6.
5. Canese L, Cardarilli GC, Di Nunzio L, Fazzolari R, Giardino D, Re M, et al. Multi-agent reinforcement learning: a review of challenges and applications. *Appl Sci*. 2021;11(11):4948.
6. Ding H, Xu Y, Hao BCS, Li Q, Lentzakis A. A safe reinforcement learning approach for multi-energy management of smart home. *Elect Power Syst Res*. 2022;210:108120. doi:10.1016/j.epsr.2022.108120.
7. Sabegh M. Deep reinforcement learning and model predictive control approaches for the scheduled operation of domestic refrigerators. Lincoln, UK: University of Lincoln; 2022.
8. Rasras M, Marin I, Radu S. Smart home environment modelled with a multi-agent system. arXiv:2304.08494. 2023.
9. Budakova D, Dakovski L, Petrova-Dimitrova V. Smart shopping cart learning agents. *Int J Adv Internet Technol*. 2019;12(3–4):109–21.
10. Budakova DV, Dakovski LG, Petrova-Dimitrova VS. Smart shopping cart learning agents development. *IFAC-PapersOnLine*. 2019;52(25):64–9. doi:10.1016/j.ifacol.2019.12.447.
11. Borgida A, Serafini L. Distributed description logics: directed domain correspondences in federated information sources. In: OTM Confederated International Conferences “On the Move to Meaningful Internet Systems”. Berlin/Heidelberg, Germany: Springer; 2002. p. 36–53.
12. Haque HMU, Khan SU, Hussain I. Semantic knowledge transformation for context-aware heterogeneous formalisms. *Int J Adv Comput Sci Appl*. 2019;10(12):664–70. doi:10.14569/ijacsa.2019.0101285.
13. Haque HMU, Khan SU. A context-aware reasoning framework for heterogeneous systems. In: 2018 International Conference on Advancements in Computational Sciences (ICACS); 2018 Feb 19–21; Lahore, Pakistan. p. 1–9.
14. Mahfooz Ul Haque H, Rakib A, Uddin I. Modelling and reasoning about context-aware agents over heterogeneous knowledge sources. In: Context-Aware Systems and Applications: 5th International Conference, ICCASA 2016; 2016 Nov 24–25; Thu Dau Mot, Vietnam. Cham, Switzerland: Springer; 2017. p. 1–11.
15. Serafini L, Tamilin A. Drago: distributed reasoning architecture for the semantic web. In: European Semantic Web Conference. Berlin/Heidelberg, Germany: Springer; 2005. p. 361–76.
16. Zhang D, Gu T, Wang X. Enabling context-aware smart home with semantic web technologies. *Int J Human-Friendly Welfare Robotic Syst*. 2005;6(4):12–20.
17. Uddin I, Rakib A, Ali M, Mahfooz Ul Haque H, Uddin A. A semantic-based approach to modelling smart indoor kitchen garden. *Mob Netw Appl*. 2024;29(4):1295–310.
18. Gruber TR. A translation approach to portable ontology specifications. *Knowl Acquis*. 1993;5(2):199–220.
19. Grau BC, Horrocks I, Motik B, Parsia B, Patel-Schneider P, Sattler U. OWL 2: the next step for OWL. *J Web Semant*. 2008;6(4):309–22.
20. Cao Y, Zhao H, Cheng Y, Shu T, Chen Y, Liu G, et al. Survey on large language model-enhanced reinforcement learning: concept, taxonomy, and methods. *IEEE Trans Neural Netw Learn Syst*. 2025;36(6):9737–57. doi:10.1109/tnnls.2024.3497992.
21. Liu W, Dong L, Liu J, Sun C. Knowledge transfer in multi-agent reinforcement learning with incremental number of agents. *J Syst Eng Electron*. 2022;33(2):447–60. doi:10.23919/jsee.2022.000045.
22. Iftikhar A, Ghazanfar MA, Ayub M, Alahmari SA, Qazi N, Wall J. A reinforcement learning recommender system using Bi-clustering and Markov decision process. *Expert Syst Appl*. 2024;237:121541. doi:10.5220/0011300300003277.
23. Hu S, Hady MA, Qiao J, Cao J, Pratama M, Kowalczyk R. Adaptability in multi-agent reinforcement learning: a framework and unified review. arXiv:2507.10142. 2025.
24. Hady MA, Hu S, Pratama M, Cao J, Kowalczyk R. Multi-agent reinforcement learning for resources allocation optimization: a survey. arXiv:2504.21048. 2025.
25. Bahrpeyma F, Reichelt D. A review of the applications of multi-agent reinforcement learning in smart factories. *Front Rob AI*. 2022;9:1027340. doi:10.3389/frobt.2022.1027340.
26. Suman S, Etemad A, Rivest F. Potential impacts of smart homes on human behavior: a reinforcement learning approach. *IEEE Trans Artif Intell*. 2021;3(4):567–80. doi:10.1109/tai.2021.3127483.

27. Yu L, Xie W, Xie D, Zou Y, Zhang D, Sun Z, et al. Deep reinforcement learning for smart home energy management. *IEEE Internet Things J.* 2019;7(4):2751–62. doi:10.1109/jiot.2019.2957289.
28. Kazmi H, Suykens J, Balint A, Driesen J. Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads. *Appl Energy.* 2019;238:1022–35.
29. Lu J, Mannion P, Mason K. A multi-objective multi-agent deep reinforcement learning approach to residential appliance scheduling. *IET Smart Grid.* 2022;5(4):260–80.
30. Huang S, Sun C, Gong J, Pompili D. Reinforcement learning-based task allocation and path-finding in multi-robot systems under environment uncertainty. *Comput Aided Civ Infrastruct Eng.* 2025;40(22):3408–29. doi:10.1111/mice.13535.
31. Pan J, Wang T, Claudel C, Shi J. Trust-MARL: trust-based multi-agent reinforcement learning framework for cooperative on-ramp merging control in heterogeneous traffic flow. *arXiv:2506.12600.* 2025.
32. Zhong Y, Kuba JG, Feng X, Hu S, Ji J, Yang Y. Heterogeneous-agent reinforcement learning. *J Mach Learn Res.* 2024;25(32):1–67.
33. Heik D, Bahrpeyma F, Reichelt D. Adaptive manufacturing: dynamic resource allocation using multi-agent reinforcement learning. In: *Proceedings of the 20th AALE Conference; 2024 Mar 6–8; Bielefeld, Germany.*