Security of Infocommunication Systems and Internet of Things

2026 Vol 4, No 1

https://doi.org/10.31861/sisiot2026.1.01006

Received 10 March 2026; revised 07 April 2026; accepted 09 June 2026; published 30 June 2026

Rule-Based and Reinforcement Learning Approaches in Multi-Agent Systems for Indoor Microclimate Control

Roman Mysiuk* and Stepan Bautin

System Design Department, Faculty of Electronics and Computer Technologies, Ivan Franko National University of Lviv, Lviv, Ukraine

*Corresponding author (E-mail: roman.mysiuk@lnu.edu.ua)

ABSTRACT This paper presents the design, implementation, and experimental validation of an edge-based agentic artificial intelligence system for adaptive smart home control. Unlike conventional rule-based automation, which relies on fixed threshold logic and predefined responses, the proposed approach models the home environment as a decision-making process in which an autonomous agent learns optimal control strategies through interaction with its surroundings. The system is deployed on a resource-constrained microcontroller platform and operates without reliance on cloud infrastructure, ensuring low latency, enhanced privacy, and reduced network dependency. Environmental parameters including temperature, humidity, illumination level, and weather-related signals are continuously monitored and encoded into a discrete state representation. Based on these states, the agent selects control actions for fan motor and LED lighting using a reinforcement learning strategy. The reward mechanism integrates comfort deviation and energy consumption into a unified objective function, enabling goal-directed behavior rather than reactive switching. The learning process gradually enables the agents to form stable control policies, resulting in convergence of environmental parameters toward their optimal values. Experimental evaluation conducted over a rule based and reinforcement learning approaches and extended condition based on microclimate parameters. The findings show that the agents progressively learn optimal control strategies, resulting in a compact clustering of environmental states around target values and demonstrating robust performance despite possible sensor noise and random action exploration. The proposed architecture illustrates how embedded reinforcement learning can transform traditional automation into a self-adaptive, memory-driven intelligent agent suitable for real-time smart home environments. The study confirms the feasibility of implementing agentic artificial intelligence on edge device ESP32 connected with sensors in porotype and highlights its potential for scalable, autonomous building control applications.

KEYWORDS smart home control, edge artificial intelligence, reinforcement learning, multi agent system, Internet of Things.

I. INTRODUCTION

The rapid growth of smart home technologies has intensified the demand for intelligent control systems capable of maintaining indoor comfort while minimizing energy consumption. Modern residential environments increasingly integrate distributed sensors, wireless communication modules, and embedded controllers to regulate ventilation, lighting, and climate parameters. However, most commercially deployed smart home solutions still rely on deterministic rule-based logic, where predefined thresholds trigger specific actuator responses. Although such systems are simple, predictable, and computationally inexpensive, they lack adaptability and are unable to optimize long-term performance under dynamic environmental conditions.

Variations in occupancy patterns, seasonal changes, and stochastic fluctuations in temperature or humidity often require manual recalibration of thresholds, limiting the scalability and robustness of conventional automation. Recent advances in artificial intelligence have opened new possibilities for adaptive building control. In particular, reinforcement learning enables systems to learn optimal behavior through continuous interaction with their environment rather than relying on static rules. Instead of reacting solely to current measurements, a learning-based controller evaluates the long-term consequences of its actions and progressively refines its decision-making policy. This paradigm shift transforms a reactive control mechanism into a goal-directed autonomous agent capable of balancing multiple objectives, such as thermal comfort and energy efficiency. Despite its theoretical advantages, the practical deployment of reinforcement learning in embedded smart home systems remains challenging due to computational constraints, limited memory, and the need for stable real-time operation without cloud dependence.

Edge computing provides a promising solution to these limitations. By executing learning and decision-making algorithms directly on microcontroller-based platforms, edge systems reduce latency, enhance data privacy, and eliminate reliance on continuous network connectivity. However, implementing adaptive artificial intelligence on resource-constrained devices requires careful state representation, algorithm selection, and memory management. Tabular reinforcement learning methods offer a lightweight alternative to computationally intensive deep learning approaches, making them suitable for low-power embedded environments. This study proposes an edge-based agentic artificial intelligence system for smart home control that integrates reinforcement learning within a microcontroller architecture. The system models environmental regulation as a sequential decision-making problem in which an autonomous agent observes sensor data, selects control actions for ventilation and lighting, evaluates outcomes through a reward function, and updates its internal policy accordingly. The reward formulation explicitly encodes a trade-off between user comfort and energy consumption, enabling systematic optimization rather than heuristic tuning.

The primary contributions of this work are as follows. First, a formal agent-based control model is developed using a structured decision framework suitable for embedded implementation. Second, a lightweight reinforcement learning algorithm is designed to operate within strict hardware limitations while maintaining stability and convergence. Third, a comprehensive experimental comparison with example on prototype of smart house. Finally, the study demonstrates that adaptive agentic behavior can be achieved entirely at the edge, without external computational infrastructure.

II. ANALYSIS OF RECENT RESEARCH AND PUBLICATIONS

The analysis of modern research in the field of Internet of Things (IoT) systems and multi-agent technologies shows the active development of solutions based on embedded platforms and edge computing [1]. In the works on the development of lightweight IoT gateways based on ESP32, the effectiveness of using microcontrollers for monitoring and local data processing in smart home systems has been confirmed [1, 2]. Studies on intelligent automation systems based on ESP32 demonstrate the possibility of integrating sensor data with automatic control mechanisms [3-5].

In most practice-oriented works, the main attention is paid to the architecture of devices, power consumption and the implementation of basic control algorithms, but there is no deep adaptive logic of decision-making [6, 7].

A few works [5, 8] propose complex automation systems with monitoring functions, but they do not take into account the interaction of several autonomous agents.

Research in the field of automated security and control systems based on Internet of Things platforms shows the possibility of integrating cloud services and sensor modules, but solutions are based on threshold rules without the use of reinforcement learning [9, 10].

Multi-agent approaches are used in the tasks of intelligent control of lighting and distributed resource management systems, which ensures decentralization of decision-making [11]. However, in such systems, the behavior of agents is mostly determined by predetermined rules, without self-learning mechanisms [11, 12]. The concept of digital twins in a multi-agent environment demonstrates the scalability and flexibility of the architecture, but the issue of adaptive optimization of parameters in real time remains open [12]. Approaches to the creation of explainable agents in IoT systems increase the transparency of decision-making, but do not provide automatic adjustment of parameters based on the experience of interaction with the environment [13].

The literature analysis also shows that intelligent monitoring and emergency response systems work effectively in real time, but the integration of reinforcement learning algorithms in such solutions is practically not considered [14].

Reinforcement Learning has recently been explored for controlling complex energy systems using surrogate models and techniques such as Double Deep Q-Networks (DDQN) for efficient training with limited data [15], while the integration of IoT technologies with artificial intelligence enables smart environments to interpret contextual information and adapt services through multi-agent systems that negotiate service settings based on user preferences [16].

III. RESEARCH RESULTS

The results of the experimental implementation of the smart house prototype are presented in this section. The system is tested using multiple environmental sensors connected to the ESP32 microcontroller to evaluate a multi-agent system with a reinforcement learning approach and a rule-based control approach [17].

A. Edge-based settings.

The experimental setup (Fig. 1) consisted of an ESP-WROOM-32 microcontroller interfaced with five LEDs, each connected to a dedicated GPIO pin through a resistor, a light-dependent resistor (LDR) for ambient light measurement, an OLED display for real-time visualization, a DHT11 sensor for temperature and humidity monitoring, a rain sensor, a small DC motor acting as a ventilator, and the necessary connecting wires for signal and power distribution.

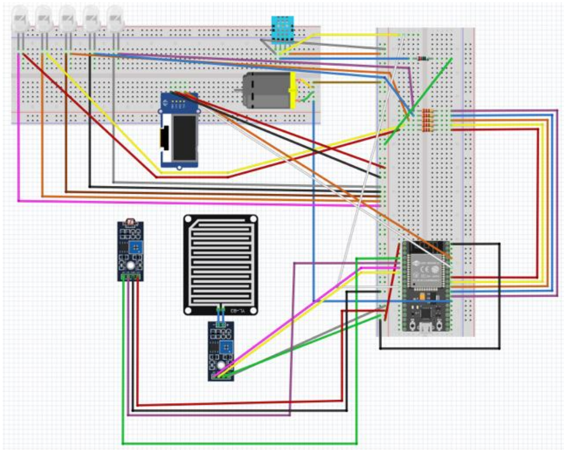

FIG. 1. Schematic of the hardware part of the project modeled in Fritzing.

Fig. 2 shows a prototype of a smart house implemented using an ESP32 microcontroller and environmental sensors, demonstrating a small-scale IoT system for automated monitoring and control of home conditions.

FIG. 2. Prototype of smart house based on ESP32 and sensors.

Fig. 3 presents the wiring scheme of the prototype, illustrating the connections between the ESP32 microcontroller and the integrated sensors used for data collection and control within the smart house system.

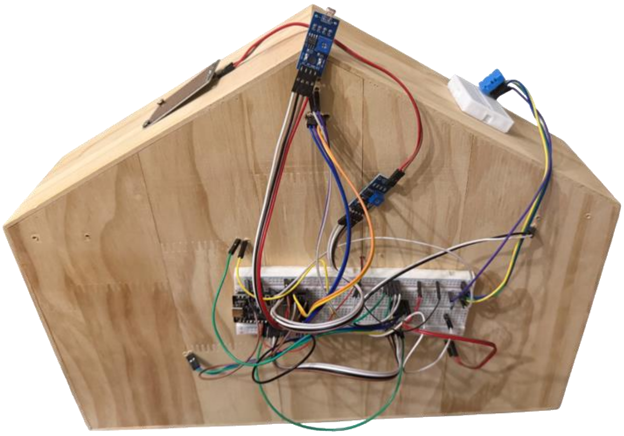

FIG. 3. Schema of connected all sensors and ESP32.

B. Client-side part. The client-side software was implemented using the React library, which is based on a component-oriented architecture widely adopted for developing high-performance and scalable single-page applications. Within this paradigm, user interface elements are structured as reusable and hierarchical components that encapsulate their own logic, styling, and behavior, while communication between components is handled through properties and efficient rendering is ensured by the virtual Document Object Model mechanism. The project was developed within a modern ecosystem using the Vite build tool, TypeScript for static typing and improved code reliability, Tailwind CSS for utility-based styling. A modular design strategy was applied, separating communication logic into a custom hook responsible for Bluetooth connectivity and state management, while container and presentation components were organized to isolate data handling, device control, and sensor visualization, thereby enhancing maintainability, testability, and scalability of the overall system architecture.

The client-side manages the full Web Bluetooth interface, including GATT server connections, services, and thirteen characteristics with unique UUIDs. The hook uses reactive state for device and sensor data and persistent references for non-reactive objects, ensuring stable, real-time access throughout the session. Device initialization is performed asynchronously, with parallel service discovery, characteristic setup, notifications activation, and initial value synchronization. Event handlers provide immediate UI updates on sensor or actuator changes, while dedicated write functions toggle device states reliably. Additional mode-control functions allow seamless switching between manual and automated operation under ESP32 control.

The implemented architecture (Fig. 4) of the web application is built on a component principle and includes several logical levels: communication, container, and display/control layer. The communication layer is implemented through a custom React hook useBluetooth.ts, which provides connection to the ESP32 Bluetooth Low Energy (BLE) server, processing asynchronous messages, and sending commands to device characteristics. The container App.tsx component coordinates the operation of all modules and passes states and methods to child components, including ConnectionControl.tsx for BLE connection control, SensorData.tsx for sensor data visualization, LEDControl.tsx and MotorControl.tsx for remote control of actuators, and HouseControl.tsx aggregates their operation and implements the overall system logic.

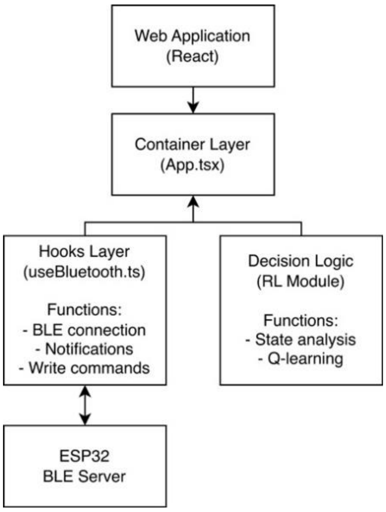

FIG. 4. Software Architecture of the Web Control System.

Additionally, the reinforcement learning unit uses sensory data to form optimal control actions, transmitting parameterized commands to ESP32. This architecture provides modularity, clear separation of responsibilities, integration of intelligent algorithms and the ability to scale and expand the functionality of the system.

Fig. 5 illustrates the web-based interface of the smart home prototype used for monitoring and controlling environmental parameters. The interface demonstrates a successful connection to the ESP32 device via BLE and displays real-time sensor data, including temperature, humidity, light intensity, and fan status.

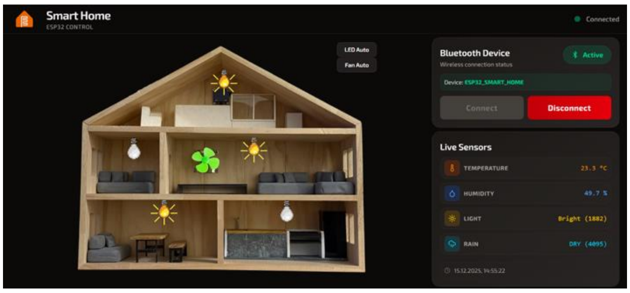

FIG. 5. Web interface with connected ESP32 device via BLE.

B. Rule-based settings. To evaluate the automatic lighting control mechanism, the system was tested under varying illumination conditions. When the ambient light level was sufficient (“Bright”), the LEDs remained off, while under low-light conditions (“Dark”), the LEDs were automatically activated, demonstrating the system’s ability to respond in real time to sensor measurements. Similarly, the automatic motor control was evaluated under two climatic triggers elevated relative humidity above 70% and detection of rainfall by the rain sensor. In both cases, the motor or ventilator engaged immediately upon the corresponding condition being met, confirming that the system correctly interprets sensor inputs and executes the programmed actuation rules. Overall, these tests validated that the software reliably processes environmental data and autonomously controls the connected devices in accordance with the designed algorithms.

C. Multi-Agent system with reinforcement learning approach. The developed system (Fig. 6) is a multi-agent architecture designed to collect data from sensors, analyze the state of the environment and automatically control actuators. The user's interaction with the system is carried out through a web platform (Human/Web Agent), with the help of which you can send control commands, change the rules of operation and receive telemetry data. Commands are transmitted via a BLE wireless channel to the ESP32 microcontroller, which acts as a coordinating agent. Sensor agents monitor environmental parameters in such a way that the lighting agent based on LDR determines the level of illumination, the climatic agent with DHT11 measures temperature and humidity, and the weather agent (rain sensor) records the presence of precipitation. The obtained data are combined into a single description of the state of the environment. Based on the formed state of the environment, the decision-making agent determines the further actions of the system.

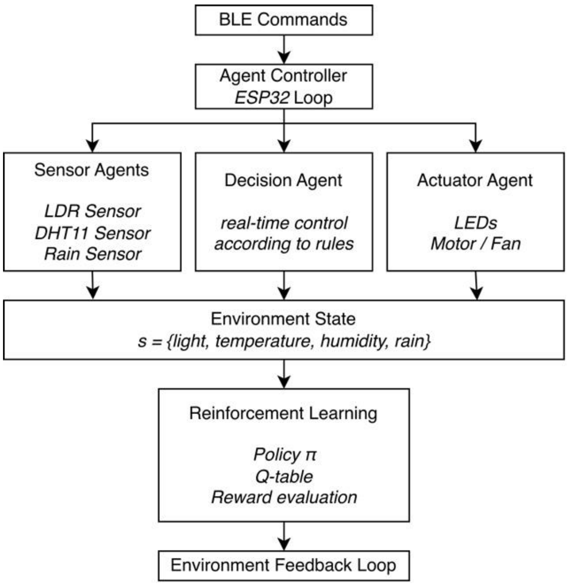

FIG. 6. Multi-Agent IoT Control Architecture.

For this, two approaches can be used: rule-oriented mechanism or reinforcement learning methods.

In the first case, decisions are made in accordance with predetermined rules (for example, turning on the lights at low light levels or activating the fan at high humidity).

In the second case, a reinforcement learning agent is used, which analyses the state of the environment and chooses the optimal action according to the learned policy, applying the Q-learning algorithm and the reward function. The selected actions are implemented by actuator agents that control lighting (LED) and climatic devices (fan or motor). After the action is performed, the parameters of the environment change, the sensors read new values again, and feedback is formed, which creates a closed control cycle.

The model selects parameters experimentally for reinforcement learning. A learning factor of $\alpha = 0.1$ provides a gradual update of Q-values, a discount factor of $\gamma = 0.9$ takes into account the importance of future rewards, and a random action probability of $\varepsilon = 0.2$ maintains a balance between exploration and exploitation. The state of the environment is sampled into 10 levels for temperature, humidity, and illumination, allowing for efficient implementation of tabular Q-learning without loss of accuracy. The reward function of exponential penalties for deviations from optimal values (temperature 22 °C, humidity 60%, illumination 500) and additional rain penalty incentivizes agents to maintain stable environmental parameters.

Fig. 7 shows the learning curve of reinforcement learning agents in a simulated climate system with small noise and smoothly increased temperature. The x-axis shows the episode number, and the y-axis shows the total reward, which reflects the degree of approximation of the environmental states to the optimal parameters (temperature 22 °C, humidity 60%, illumination 500). At the beginning of training, lower reward values are observed due to random selection of actions by agents and lack of experience, while the growth of the curve and reaching a stable level of ~2.9 – 3.0 demonstrates the effective formation of optimal management policies. Short-term fluctuations are due to stochastic sensor noises and ε-greedy choice of actions, which provides a balance between research and operation.

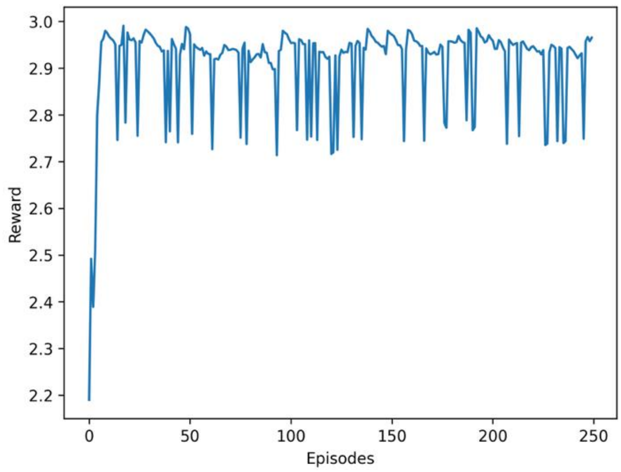

FIG. 7. Learning rewards with 250 episodes.

The trajectory of changes in the environment is shown in Fig. 8 to identify patterns and grouping of data. As a result of the simulation, a three-dimensional scatter diagram was obtained, reflecting the relationship between the values of temperature, humidity and the target indicator of the system during successive episodes of training. Analysis of the obtained data indicates the formation of a dense group of points in the temperature range of approximately from 21.8 °C to 22.3 °C and humidity from 58% to 61%, which indicates stabilization of environmental parameters or system control policies. Such a concentration of values may be due to the convergence of the learning algorithm, when the agent, in the process of optimization, finds the most efficient region of states and reproduces it in subsequent episodes. The presence of isolated distant points with higher values of temperature (≈23 – 24.5 °C) and humidity (≈64 – 69%) can be explained by the phase of exploring the state space, characteristic of the initial stages of learning, or by the influence of random disturbances of the environment. Thus, the results obtained indicate a limited variability of the main parameters of the system and a gradual stabilization of the behaviour of the model, which is manifested in the formation of a compact cluster of observations in the feature space.

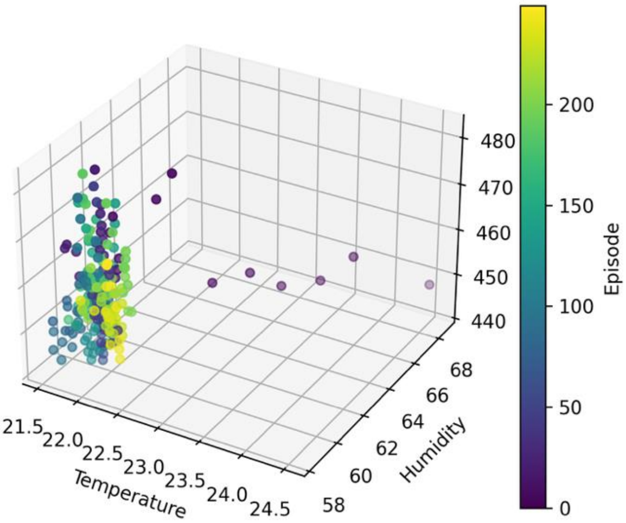

FIG. 8. 3D Environment Trajectory.

In the process of training the reinforcement algorithm, a policy for controlling the system's actuators, including lighting and ventilation, is formed. The policy determines the choice of the optimal action for each state of the environment, which is described by a combination of temperature, relative humidity and light level parameters. Based on the accumulated experience of interaction with the environment, the agent estimates the expected reward for various actions and gradually gives preference to those controlling influences that ensure the maximization of the reward function.

The lighting control policy aims to maintain the light level near the specified optimal value. In the case when the lighting level is lower than desired, the algorithm tends to choose the action of turning on the LED lighting, which leads to an increase in the light intensity in the medium. Instead, when the optimal light level is exceeded, the system can choose to turn off or decrease the lighting, which allows you to stabilize the parameter within the desired range. Thus, an adaptive lighting regulation strategy is formed, which ensures the maintenance of favourable conditions for the functioning of the system.

The fan control policy aims to regulate the temperature and relative humidity of the environment. In conditions of high humidity or temperature, the algorithm selects fan activation actions with different intensities (low or high-speed modes), which helps to reduce the corresponding parameters. In situations (Fig. 9) where temperature and humidity values are close to optimal, the system prefers to turn off the fan.

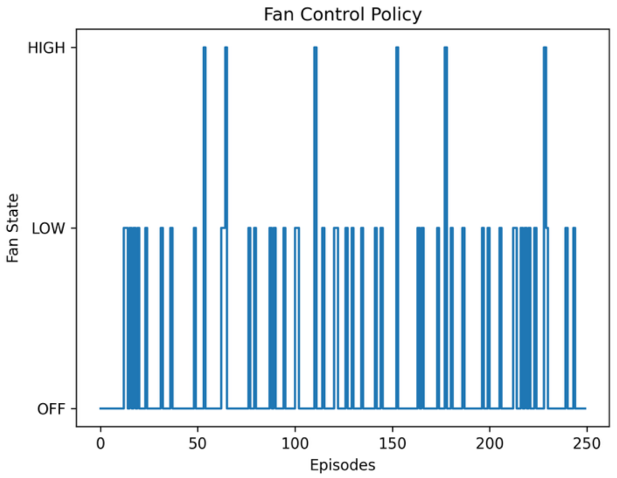

FIG. 9. Result of fan control policy.

IV. DISCUSSION

The proposed approach based on reinforcement learning demonstrates a conceptual transition from reactive automation to an autonomous agent-oriented decision-making system. In contrast to deterministic threshold logic, the RL model forms management policies by maximizing long-term cumulative rewards, which allows both the current state of the environment and the future consequences of actions to be taken into account. The reinforcement learning agent gradually adjusts the management policy without the need to manually reconfigure the thresholds. Thus, the integration of Q-learning transforms the system from a deterministic controller into a built-in rational agent with internal memory and an optimization mechanism, which allows it to be reasonably classified as an edge-based agentic AI system. At the same time, the reinforcement learning approach requires additional computational resources and a training phase, which is important for microcontroller implementations.

The learning curve (Fig. 7) demonstrates a rapid increase in reward during the initial episodes, indicating that the agents quickly learn a basic control policy for stabilising environmental parameters. After approximately 20–30 episodes, the reward values stabilize, suggesting convergence of the learning process and successful adaptation of the control strategy. Occasional drops in reward can be observed throughout the training process. These fluctuations are caused by sensor noise, stochastic environmental dynamics, and exploration actions generated by the ε-greedy policy. The learning curve indicates that convergence is achieved after approximately 30 episodes. Therefore, the total number of 250 training episodes is sufficient to ensure stable learning and policy refinement. The number of training episodes represents a tunable hyperparameter of the reinforcement learning process. Increasing the number of episodes may further refine the learned policy but also increases computational cost.

The surface demonstrates (Fig. 10) that the system evaluates the microclimate not by individual parameters, but by their joint interaction. Therefore, even small changes in temperature can have a different impact on the comfort level depending on the humidity level. This confirms the feasibility of using a multidimensional reward function to simulate optimal microclimate parameters (humidity = 60, temperature = 22, light = 500) and further use in algorithms for adaptive control of climate systems.

Although the communication technology is not the focus of this study, Bluetooth is employed in the system because it provides low-latency, energy-efficient, and direct device-to-device communication, which facilitates the coordination and responsiveness required in a multi-agent IoT architecture during experimental evaluation.

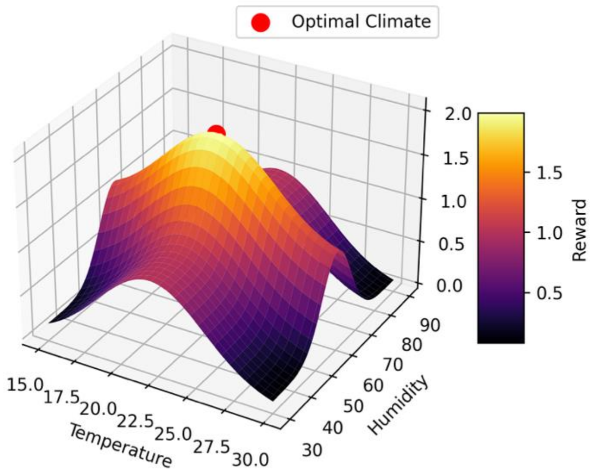

FIG. 10. 3D Temperature–Humidity Reward Surface.

As this study includes a comparison between rule-based and reinforcement learning approaches, their key characteristics are summarized in Table 1.

TABLE 1. Comparative analysis of rule-based and RL approaches.

Criterion	Rule-based	Reinforcement Learning
Policy Type	Fixed	Adaptive
Memory	No	Q-table
Optimization	No	Maximization of reward
Adaptation to environment	No	Yes
Reaction to changing conditions	Statical	Dynamical
Energy efficiency	Depends on the thresholds	Optimized
Computational complexity	Low	Moderate

A promising area of further research is the extension of the model to a multi-room multi-agent architecture and the integration of predictive models to improve the convergence rate. The proposed approach can be applied in smart homes, intelligent buildings, and IoT-based environmental control systems.

V. CONCLUSION

In conclusion, the conducted experimental modeling demonstrates that the proposed multi-agent control system is capable of effectively adapting environmental control parameters in a stochastic smart-house environment. The obtained results indicate stable convergence of the learning process and consistent improvement of the system's performance indicators during training episodes. The study confirms that reinforcement learning can successfully enhance decision-making in distributed IoT environments compared with traditional rule-based approaches, providing greater adaptability to dynamic environmental conditions. The developed architecture allows agents to coordinate their actions while operating under the computational constraints typical for embedded and edge devices.

The obtained results demonstrate stable convergence of the reinforcement learning model within 250 training episodes, achieving an average reward of about 2.9 – 3.0 and maintaining environmental parameters close to optimal values (22 °C, 60 %), with temperature stabilized in the range of 21.8 – 22.3 °C and humidity within 58 – 61%.

The practical significance of the proposed approach lies in the possibility of implementing adaptive control directly at the edge level without reliance on centralized cloud infrastructure. Such capability is particularly relevant for autonomous smart environments, including intelligent buildings, energy management systems, and distributed IoT networks requiring low-latency and energy-efficient operation. Future research may focus on extending the proposed framework to larger multi-agent systems, integrating additional environmental parameters, and evaluating the approach in real-world smart-building deployments.

AUTHOR CONTRIBUTIONS

R.M. – conceptualization, methodology, investigation, supervision; S.B. – software, validation; R.M., S.B. – writing-original draft preparation, writing-review and editing.

COMPETING INTERESTS

The authors declare no conflict of interest.

REFERENCES

[1] F. Serepas, I. Papias, K. Christakis, N. Dimitropoulos, and V. Marinakis, "Lightweight Embedded IoT Gateway for Smart Homes Based on an ESP32 Microcontroller," *Computers*, vol. 14, no. 391, 2025, doi: 10.3390/computers14090391.

[2] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and Implementation of ESP32-Based IoT Devices," *Sensors*, vol. 23, 6739, 2023, doi: 10.3390/s23156739.

[3] G. Kaur et al., "Design and Implementation of an Intelligent Home Automation System Using ESP32 and IoT Integration," in *Proc. ICDAM 2025*, Springer, *Lecture Notes in Networks and Systems*, vol. 1594, 2026, doi: 10.1007/978-3-032-02949-2_42.

[4] P. Skibel, F. Matiko, and I. Demkiv, "Remote Monitoring System for Microclimate Parameters of Smart Home and Industrial Premises Based on ESP8266 Microcontroller," *Energy Engineering and Control Systems*, vol. 11, no. 2, pp. 199–209, 2025, doi: 10.23939/jeecs2025.02.199.

[5] M. Iqbal et al., "Prototype of Automatic Security and Control System for Smart Home Using ESP32 Microcontroller," *Journal of Emerging Innovations in Engineering*, vol. 1, no. 02, pp. 54–65, 2025, doi: 10.65664/jeie.v1i02.9.

[6] M. Y. Yakubovych, V. L. Lyashkevych, and R. Y. Shuvar, "Energy Conservation as One of the Components of the

Management System for the Smart Sustainable Workspaces," *Electronics and Information Technologies*, vol. 29, pp. 79–94, 2025, doi: 10.30970/eli.29.8.
[7] O. L. Korenivska et al., "IoT Monitoring System for Microclimate Parameters in Educational Institutions Using Edge Devices," in *CEUR Workshop Proceedings*, vol. 3374, 2023. Available: https://ceur-ws.org/Vol-3374/paper05.pdf
[8] D. Sinaga et al., "Design and Implementation of an ESP32-Based IoT Smart Room System with Automated Temperature, Ventilation, and Lighting Control," *Journal of Intelligent Systems, Embedded Systems, and Computer Engineering*, vol. 1, no. 1, pp. 1–11, 2026, doi: 10.65358/rva03t27.
[9] G. Verma et al., "Smart Home Automation with Smart Security System over the Cloud," in *Proc. 14th Int. Conf. on Computing Communication and Networking Technologies (ICCCNT)*, Delhi, India, 2023, pp. 1–7, doi: 10.1109/ICCCNT56998.2023.10306548.
[10] S. Kouah et al., "Internet of Things-Based Multi-Agent System for the Control of Smart Street Lighting," *Electronics*, vol. 13, no. 3673, 2024, doi: 10.3390/electronics13183673.
[11] M. Russo et al., "Advancing Robotic Systems with Distributed Multi-Agent Digital Twins: A Scalable and Adaptive Framework," *SN Computer Science*, vol. 6, no. 975, 2025, doi: 10.1007/s42979-025-04525-w.
[12] K. M. et al., "Design and Implementation of Dual-Function System for Inventory Management and Safety: IoT-Based Monitoring and Gas Detection," in *Proc. 2025 Int. Conf. on Multi-Agent Systems for Collaborative Intelligence (ICMSCI)*, Erode, India, 2025, pp. 400–407, doi: 10.1109/ICMSCI62561.2025.10894350.
[13] H. Zhang, R. Zhang, and J. Sun, "Developing Real-Time IoT-Based Public Safety Alert and Emergency Response Systems," *Scientific Reports*, vol. 15, no. 29056, 2025, doi: 10.1038/s41598-025-13465-7.
[14] A. Dobrovolskis, E. Kazanavičius, and L. Kižauskienė, "Building XAI-Based Agents for IoT Systems," *Applied Sciences*, vol. 13, no. 4040, 2023, doi: 10.3390/app13064040.
[15] K. Mandaric, A. Keselj Dilberovic, and G. Jezic, "A Multi-Agent System for Service Provisioning in an Internet-of-Things Smart Space Based on User Preferences," *Sensors*, vol. 24, no. 1764, 2024, doi: 10.3390/s24061764.
[16] A. Stevenson, M. Panwar, R. Hovsapian, and A. Sarwat, "Efficient Reinforcement Learning for Real-Time Hardware-Based Energy System Experiments," *Proc. AAAI Symposium Series*, vol. 2, no. 1, pp. 153–158, 2024, doi: 10.1609/aaaiss.v2i1.27663.
[17] S. Bautin and R. Mysiuk, "Intelektualna systema monitorynhu klimatychnyh umov ta keruvannya prystroyamy na bazi mikrokontrolera ESP32," *Radioelectronics and Computer Technologies: Proceedings of the 2nd Ukrainian Scientific-Practical Seminar*, I. D. Karbovnyk, ed. Lviv, Ukraine: Ivan Franko National University of Lviv, 2026. Available: https://electronics.lnu.edu.ua/wp-content/uploads/RCT-2026-Abstracts.pdf (In Ukrainian)

Roman Mysiuk

PhD in Computer Science with hands-on experience in test automation, data analytics, and machine learning technologies across multiple domains. He has over 7 years of experience in the IT industry and 5 years in academia, combining practical software development with research and teaching.

ORCID ID: 0000-0002-7843-7646

Stepan Bautin

Student pursuing a bachelor's degree in Computer Science, focused on designing and implementing AI-powered information systems. Research field includes artificial intelligence, smart environments, IoT integration, and applied machine learning for real-world applications.

ORCID ID: 0009-0006-1451-9243

Підходи до навчання на основі правил та навчання з підкріпленням у багатоагентних системах для контролю мікроклімату в приміщенні

Роман Мисюк*, Степан Баутін

Кафедра системного проектування, факультет електроніки та комп'ютерних технологій, Львівський національний університет імені Івана Франка, Львів, Україна

*Автор-кореспондент (Електронна адреса: roman.mysiuk@lnu.edu.ua)

АНОТАЦІЯ У цій статті представлено процес проектування, реалізації та експериментальну перевірку агентної системи штучного інтелекту на основі периферії для адаптивного керування розумним будинком. На відміну від традиційної автоматизації на основі правил, яка використовує фіксовані пороги та попередньо визначені відповіді, запропонований підхід моделює внутрішнє середовище як процес прийняття рішень, де автономний агент вивчає оптимальні стратегії керування через взаємодію з середовищем. Система розгорнута на платформі мікроконтролера ESP32 з обмеженими ресурсами та працює незалежно від хмарної інфраструктури, що забезпечує низьку затримку, підвищену конфіденційність та знижену залежність від мережі. Параметри навколишнього середовища, включаючи температуру, вологість, рівень освітлення та погодні сигнали, постійно контролюються та кодуються в дискретне представлення стану. На основі цих станів агент обирає дії керування для вентилятора та світлодіодного освітлення, використовуючи стратегію навчання з підкріпленням. Механізм винагороди інтегрує відхилення комфорту та споживання енергії в єдину цільову функцію, що дозволяє цілеспрямовану поведінку замість реактивного перемикання. Процес навчання поступово дозволяє агентам формувати стабільні політики управління, що призводить до зближення параметрів середовища до їх оптимальних значень. Експериментальна оцінка проводилася шляхом порівняння підходів на основі правил та навчання з підкріпленням, а також у розширених умовах з урахуванням параметрів мікроклімату. Результати дослідження показують, що агенти поступово навчаються оптимальним стратегіям керування, що призводить до компактної кластеризації станів навколишнього середовища навколо цільових значень та демонструє надійну продуктивність, незважаючи на шум датчиків та дослідження випадкових дій. Запропонована архітектура ілюструє, як вбудоване навчання з підкріпленням може перетворити традиційну автоматизацію на самоадаптивного інтелектуального агента, керованого пам'яттю, придатного для роботи в реальному часі в умовах розумного будинку. Дослідження підтверджує ефективність впровадження агентного штучного інтелекту на периферійному пристрої ESP32, підключеному до датчиків у прототипі, та підкреслює його потенціал для масштабованих автономних застосувань у сфері керування розумними будівлями.

КЛЮЧОВІ СЛОВА управління розумним будинком, периферійний штучний інтелект, навчання з підкріпленням, багатоагентна система, Інтернет речей.