Smart Building Operations and Virtual Assistants Using LLM

Reachsak Ly
Myers-Lawson School of Construction,
Virginia Tech
Blacksburg, Virginia, USA
reachsak@vt.edu

Alireza Shojaei†
Myers-Lawson School of Construction,
Virginia Tech
Blacksburg, Virginia, USA
shojaei@vt.edu

Xinghua Gao
Myers-Lawson School of Construction,
Virginia Tech
Blacksburg, Virginia, USA
xinghua@vt.edu

ABSTRACT

Conventional AI-powered smart home assistants primarily function as voice-activated control systems with limited adaptability and contextual understanding. Similarly, while traditional artificial intelligence has advanced autonomous building research, it often relies on predefined rules and struggles with real-time decision-making in dynamic building environments. This paper introduces a novel Generative AI-driven framework that integrates Large Language Models (LLMs) to create a smart generative AI-based virtual assistant and an operation automation system for building infrastructure. The AI systems autonomously manage building operations by analyzing real-time occupancy patterns and adjusting environmental conditions based on predefined comfort thresholds. The proposed system also facilitates seamless human- building interaction through an LLM-powered virtual assistant. The framework is validated through a prototype implementation in a real-world building equipped with smart appliances, with evaluations focusing on the AI systems' accuracy, reliability, and scalability. The findings demonstrate that the prototype system can autonomously adjust building conditions, optimize energy usage, and provide intelligent assistance for building operation tasks.

CCS CONCEPTS

• Human-centered computing →Human-computer interaction (HCI)

KEYWORDS

Generative AI, Large Language Models, Smart Buildings, Autonomous Building Operations, Virtual Assistant

ACM Reference format:

Reachsak Ly, Alireza Shojaei and Xinghua Gao. 2025. Smart Building Operations and Virtual Assistants Using LLM. In *Companion Proceedings of the 33rd ACM Symposium on the Foundations of Software Engineering (FSE '25), June 23--27, 2025, Trondheim, Norway. ACM, New York, NY, USA, 7 pages.*

† Alireza Shojaei is the corresponding author.

FSE Companion '25, June 23--27, 2025, Trondheim, Norway

1 Introduction

Research on autonomous buildings has become a promising frontier in the field of smart and sustainable infrastructure. Autonomous buildings are characterized by their ability to operate independently through self-management, self-sufficiency, and intelligent operation. Machine learning techniques and AI have played a pivotal role in enabling these advancements. Their capacity to analyze large volumes of data, recognize trends, and make well-informed choices has contributed significantly to improving energy efficiency, enhancing occupant comfort, and optimizing building performance. However, the functionality of this conventional machine learning model may rely on predefined rules or specific training data, which, in some circumstances, may not adequately capture the dynamic and complex nature of building operations. Furthermore, they often struggle to adapt to evolving circumstances or to incorporate contextual information effectively.

The advent of large language models (LLMs) offers a promising avenue to overcome these limitations and unlock a new realm of possibilities for autonomous building operations. LLMs exhibit remarkable capabilities in natural language processing, which could enable seamless human- machine interactions and intelligent decision-making processes. Unlike conventional AI models, LLMs possess a deep understanding of contextual information and can engage in humanlike conversations, allowing for more intuitive and adaptive control of building systems. Despite its transformative potential, the application of LLMs in building automation and smart virtual assistants remains largely unexplored. This study aims to address these gaps by proposing a novel Generative AI-driven framework that seamlessly integrates LLMs into building automation systems for autonomous building management while providing an intelligent virtual assistant interface for enhanced human-building interaction.

1 Related work

Several studies have been conducted to explore the application of LLM in various stages of the construction project lifecycle, including project planning, construction, operations, and maintenance. For instance, Prieto et al. [1] examines the use of LLM in construction schedule generation based on the project scopes and requirements. Within the construction phase, researchers have also explored the use of LLM in the construction

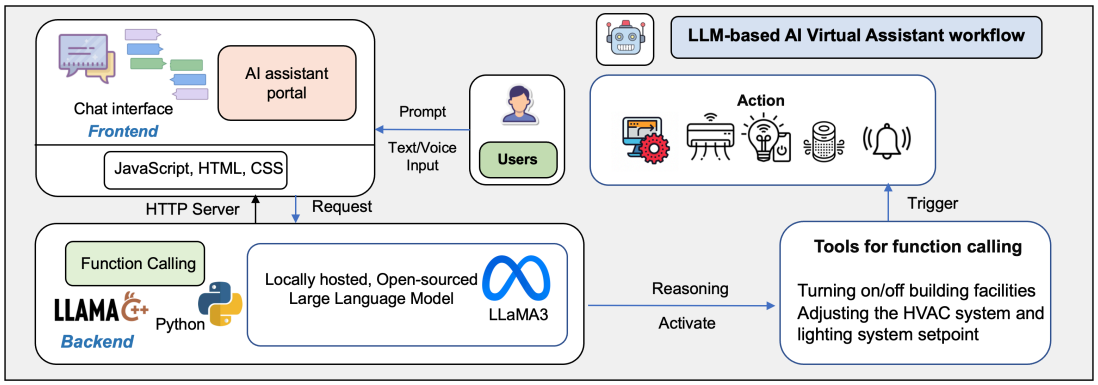

Figure 1: Overview of the LLM-based Virtual Assistant

robotics domain. You et al. [2] introduced RoboGPT, a novel system that utilizes the reasoning abilities of ChatGPT for automating sequence planning in robot-assisted assembly for construction tasks. Chen et al. [3] proposed a construction safety query system, which combines image captioning with a visual question-answering capability on head-mounted AR devices by leveraging vision language models such as ChatGPT 4. In another study, Uddin et al. [4] examined the impact of integrating ChatGPT into the construction education curriculum. The investigation involved measuring students' hazard recognition abilities before and after introducing ChatGPT as an educational tool. Zheng and Fischer [5] introduced an AI-powered virtual assistant system that integrates ChatGPT for supporting natural language-based building information model (BIMs) search. This system allows users to query BIM databases using natural language, extract relevant information, and receive responses along with 3D visualizations. While previous research has investigated the integration of GPT models in various domains of construction, the application of LLMs in the context of smart building infrastructure remains relatively unexplored. Furthermore, it is important to note that the existing research on the application of LLMs in the construction domain has primarily utilized commercialized GPT models, such as OpenAI's ChatGPT. The use of these cloud-based GPT models requires sensitive data to be transmitted to external servers for processing, potentially exposing it to unauthorized access or data breaches [6]. Another significant challenge lies in the cost and scalability aspects of these commercial LLM services. As an example, users must pay a monthly subscription fee or recurring charges based on their level of usage. Therefore, there is also a need to explore alternative solutions, such as utilizing local and open-source LLMs that can improve inference speed while addressing data privacy concerns, as the data remains localized within the device or system.

3 Methodology

This research employs the Design Science Research (DSR) [7] methodology to develop and validate the LLM-powered building automation system and virtual assistant. Through problem identification, the literature review reveals significant gaps in current building automation systems, including limited contextual understanding in traditional AI approaches and insufficient research on LLM integration in smart building applications. To address these gaps, the research objectives focus on developing an innovative framework that integrates LLMs into building automation systems for enhanced human-building interaction and autonomous environmental control. The design and development phase encompasses creating the LLM-powered virtual assistant interface, developing the autonomous building control system, and implementing real-time monitoring capabilities. The system will be demonstrated in a real-world smart building environment equipped with various sensors and smart appliances. Evaluation will assess the system's performance through multiple metrics, including environmental control accuracy, system reliability, and scalability. Finally, the research findings and framework design will be disseminated through academic publications.

4 Proposed framework

4.1 LLM-based Virtual Assistant

The LLM-based virtual assistant aims to facilitate the human-building interaction aspect within the proposed framework. Users can communicate with the virtual assistant through text and voice input to control various building facilities, adjust set points for the specific building smart facilities, or turn systems on or off as needed. Central to the virtual assistant is a locally hosted, open-source LLM and its function calling capabilities, specifically the LLaMA3 model by Meta [8]. The use of local and open-source LLM is driven by several factors, including enhanced data privacy and reduced operational costs. By keeping all interactions and data processing within the local infrastructure, the system also ensures independence from third-party entities and maintains strict control over sensitive information. The virtual assistant workflow is illustrated in Figure 1, which demonstrates the integration of frontend and backend components. Users interact with the system through a web-based chat interface. The submitted prompts by the user are transferred to the backend and processed by the local LLM model. Upon receiving a query, the LLMs model leverages its reasoning and function-calling capabilities to understand the user's request and activates the appropriate predefined tools for task execution.

4.2 Autonomous building operation

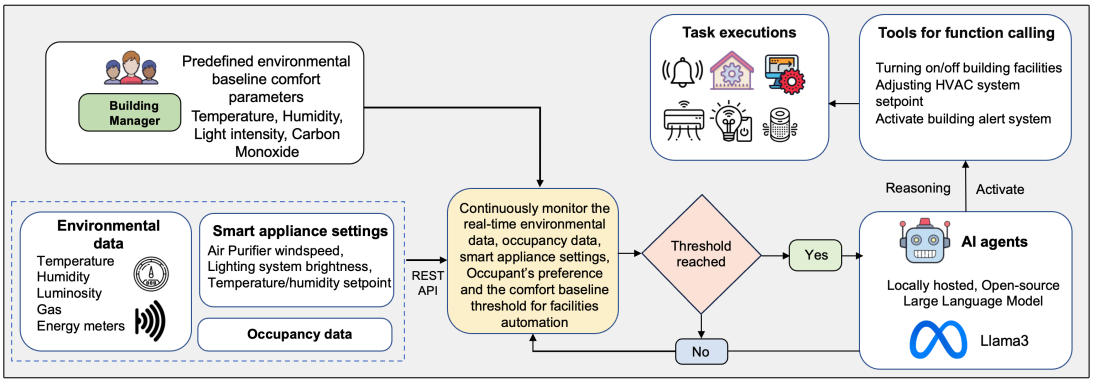

Figure 2: Overview of the LLM-powered autonomous building operation.

The proposed LLM-based autonomous building operation framework leverages the integration of IoT devices and sensors, smart building facilities such as the HVAC and lighting system, LLMs, and blockchain technologies to create automated control of building systems (Figure 2). An array of environmental sensors and IoT Devices will be used to collect environmental data such as temperature, humidity, light intensity levels, carbon monoxide levels, as well as the energy usage information and occupancy level within the physical space. Raspberry Pi devices are used to process these data and feed them into the AI agent through REST API. In addition, the threshold parameters are used for the automated smart building operation (e.g., max or min temperature, humidity, etc.). These threshold values are the baseline comfort parameters for ensuring optimal comfort and indoor environmental quality and are defined by users or building managers, who serve as administrators for the physical spaces. During the operation, the LLM-based AI agent will continuously compare real-time sensor data from the physical space against these predefined thresholds. When the environmental data values exceed these thresholds, the AI agent utilizes its function-calling capabilities to control various building systems, including smart lighting and HVAC systems. These operations involve adjusting set points, activating or deactivating devices, or triggering alerts to maintain optimal building conditions.

5 Proof of Concept

5.1 Implementation preparation

5.1.1 Smart home appliance. To simulate the required environmental management capabilities for the study, a range of smart home devices has been installed, allowing the virtual assistant to control air quality, humidity, lighting, and temperature within the physical space. To simulate air quality control within the space, this study uses a smart air purifier device, Xiaomi Smart Air Purifier 4 Compact, which is equipped with multiple fan speed configurations. In addition, a smart humidifier, the Govee Smart Humidifier H7141, is used to facilitate the adjustment of humidity levels within the room. It offers multiple fan speed settings. The Xiaomi Mi Smart Standing Fan 2 is also integrated into the system by offering multiple fan speeds for personalized airflow control. This device allows the AI system to demonstrate its ability to regulate air circulation within the space, simulating traditional HVAC systems. For lighting control, this study used a smart bulb, Yeelight Smart Light Bulbs W3. It offers adjustable brightness levels, which mimic the control of indoor lighting conditions.

5.1.2 Deployment of the AI systems module. The proposed AI systems in this study are powered by LLaMA 3 8b, an open-source LLM with 8 billion parameters. The workstation used for the LLaMA3 model deployment in this study is an Apple MacBook Pro with an M1 Max chip and 32GB of RAM. To run the LLaMA 3 model efficiently, this study employs a quantized version of the model using the llama.cpp [9]. Llama cpp is a tool that allows the execution of quantized LLMs on local hardware with support for different types of GPU. Quantization is an essential operation for the reduction of the model's size and computational requirements, which is particularly important for deploying the AI systems on local hardware [10]. This enables running complex models on devices with less computational power without severely compromising performance. The 8-bit quantized model of LLaMA 3 8b is used in this study. Function calling is one of the core features of the proposed virtual assistant. In this context, function calling refers to the AI's ability to analyze user requests, extract key information, and invoke predefined tools or functions to perform tasks. This study leverages llama-cpp-agent, a Python-based package, to implement the function-calling capabilities of the proposed virtual assistant.

5.2 Experiments

The first experiment aims to validate the LLM-based AI agent's ability to control smart building appliances. For this demonstration, the control of a smart light bulb was selected. Users can interact with the virtual assistant to issue voice commands, such as turning the light on or off. The user's prompts are then transmitted to the backend, where they are processed by the locally hosted LLMs. Upon receiving the request, the AI uses its function-calling capabilities to understand and act upon the user's instructions. It then activates a Python script that interfaces with the device's API, enabling it to control the smart light bulb as requested. In addition, instead of the simple on/off commands, users can also provide the context or hint, such as stating that the room is "too dark" or "too bright." The virtual assistant,

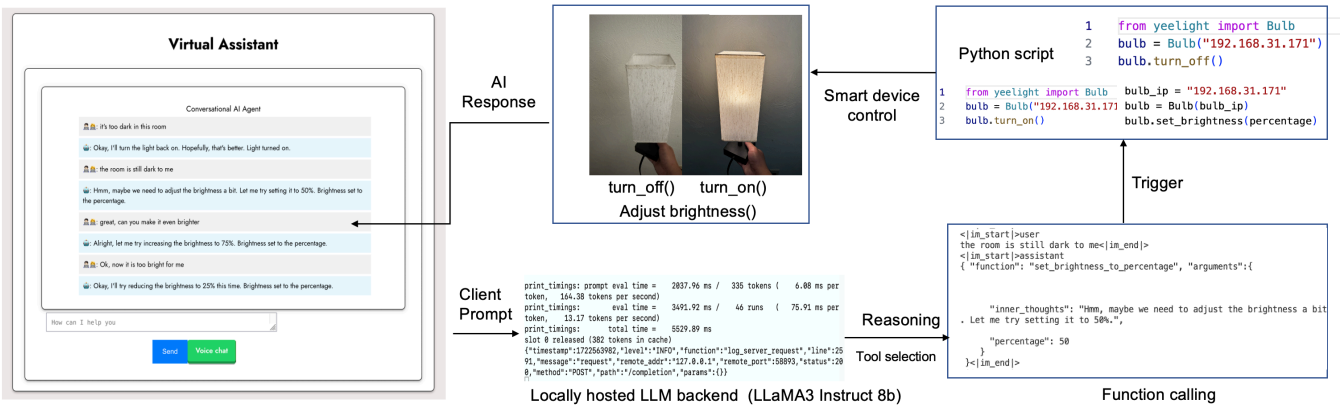

Figure 3: Smart building appliance control using LLM-based Virtual Assistant.

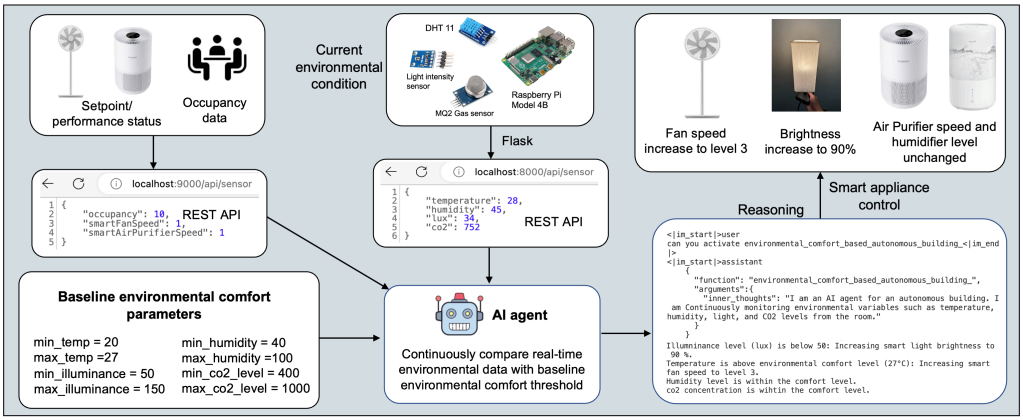

Figure 4: Autonomous smart appliance control using LLM-based AI agent.

leveraging its context-awareness capabilities, can autonomously adjust the brightness level of the light bulb, which can be seen in Figure 3. This dynamic interaction demonstrates the enhanced functionality of the LLM-powered virtual assistant, distinguishing it from traditional AI systems that lack such nuanced environmental awareness.

The second experiment explores the role of the AI agent in managing smart devices for autonomous building operations through two distinct scenarios: automatic appliance control based on occupancy and autonomous adjustments according to environmental data. In the first case, this experiment uses the simulated occupancy data (randomly selected between 1 and 20) and is updated every 1 minute to test the AI agent's ability to autonomously control devices such as a smart fan and air purifier. For demonstration, this study defines low occupancy as fewer than five individuals, medium occupancy as five to nine individuals, and high occupancy as ten or more individuals. As demonstrated in Figure 4, the AI agent continuously monitors the fan and air purifier settings and occupancy data through RESTful API. Depending on the occupancy levels, the AI modifies the performance of the devices accordingly. If no occupants are detected, the AI will turn off all devices. In the case of low occupancy, it reduces the performance to a lower setting, while in the medium occupancy case, it sets the smart appliance to medium settings. Finally, in high occupancy scenarios, it increases the devices' performance to maximum setting. Initially, both the smart fan and air purifier were set to low performance at level 1. During the experiment, the simulated occupancy was ten individuals, which led the AI to perform the contextual reasoning and raise the performance settings of the fan and air purifier to level 3 and level 7, respectively, to ensure environmental comfort in the physical space.

The second case aims to validate the AI agent's ability to autonomously adjust smart appliances based on baseline environmental comfort parameters. In this scenario, the smart appliance setpoints were initially configured at their lowest levels. As shown in Figure 4, the AI retrieved the baseline comfort parameters, which include temperature (20°C-27°C), humidity (40%-100%), light intensity (50-150 lux), and carbon monoxide levels (0-50 ppm). The real-time room conditions of 28°C, 45% humidity, 34 lux, and 752 ppm were obtained through REST API. Upon processing this data, the AI agent determined that the temperature and luminance were outside the comfort range and automatically adjusted the smart fan to speed level 3 and the smart light to 90% brightness while leaving other appliance settings unchanged. This experiment demonstrates the AI agent's ability to autonomously regulate the indoor environment by adjusting smart devices based on real-time data and predefined comfort thresholds. The code for the technical implementation of the prototypes is available under an open-source license [11][12].

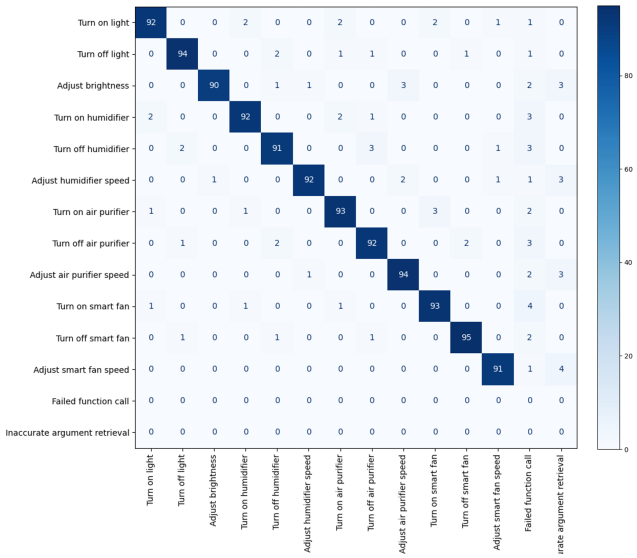

Figure 5: Confusion Matrix for Building Operations Using Virtual Assistant

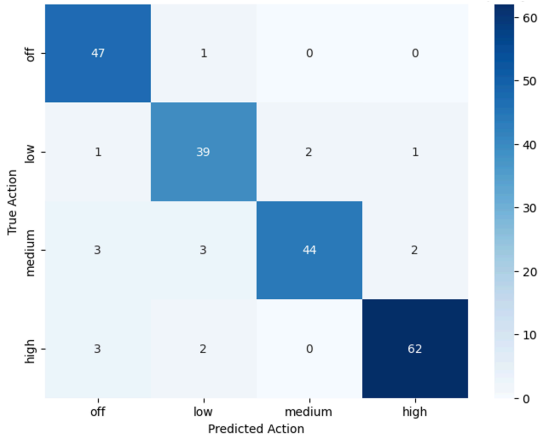

Figure 6: Confusion Matrix for Occupancy-based Autonomous Building Operations.

6 Evaluation

The LLM-based AI systems in this study are composed of two components: a Virtual assistant and an AI agent. The virtual assistants can take user commands and perform smart building tasks and blockchain tasks through activating tools using function calls. Additionally, the AI agent will monitor the environmental condition occupancy level and performance of smart building facilities control using function calls. The performance of the AI systems will be evaluated on the speed, accuracy, and reliability of the function calling capability of the AI systems. To evaluate the accuracy of the functions or tasks executed by the AI systems, this study uses five different metrics, including Precision, Recall, F1 Score, Overall Accuracy, and Reliability. These metrics are specifically useful for evaluating the model's performance when it successfully selects a function which is similar to the multiclass classification task. The reliability metric aims to examine how consistent the AI is in executing the function when responding to the same prompt.

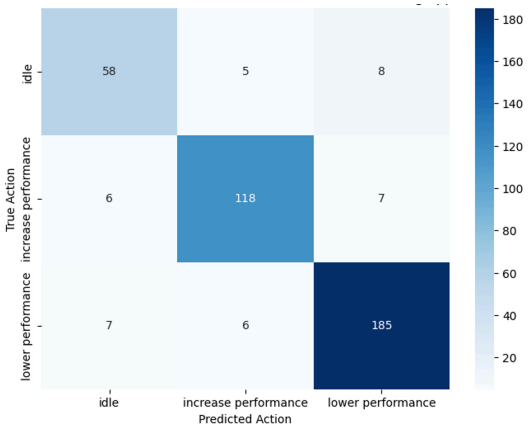

Figure 7: Confusion Matrix for Threshold-based Autonomous Building Operations

6.1 Evaluation of the Virtual assistant

There are 12 different functions/tools that will be used by the virtual assistant upon its reasoning of the user command. Those functions are Turning on the light, turning off the light, adjusting brightness, turning on and off the humidifier, Adjust the humidifier speed, turning on the air purifier, turning off the air purifier, Adjust air purifier speed, Turning on the smart fan, Turning off the smart fan, Adjust smart fan speed. Among the 12 functions, 4 of them, which are Adjust Brightness, Adjust Humidifier Speed, Adjust Air Purifier Speed, and Adjust Smart Fan Speed, will retrieve arguments (e.g., level of fan speed, etc.) from the user command. To demonstrate the variety of user requests, 10 different request prompts were used for each of the 12 functions. For the reliability metric, this study will ask the virtual assistant 10 times for the same prompt within each function. The confusion matrix for building operation with a virtual assistant for each function is shown in Figure 5. It should be noted that there are two additional columns for failed function call and Inaccurate argument retrieval, which aim to demonstrate the times when the AI systems can't understand the command of the user, error, or incorrect retrieval of argument. Additionally, to measure the task execution speed of the AI systems, this study will use tools like Llamacpp to measure the throughput and times needed to execute different tasks.

6.1 Evaluation of the AI Agent

To evaluate the AI agent's performance in automating smart building operations, two experimental setups are used: occupancy-based automation and threshold-based automation. For occupancy-based control, we define a predefined range of occupants (x_true) (e.g., 0-20 people) and corresponding smart appliance setpoints (y_true) (off, low, medium, high) as mentioned in section 5.2. The AI agent is tested with 200

Table 1: Evaluation of the proposed AI systems

AI Systems	Task	Precision (%)	Recall (%)	F1 Score (%)	Accuracy (%)	Reliability (%)
AI Assistants	Smart appliance control task	95.91	92.75	94.17	92.75	95.45
AI Agents	Occupancy based automation	90.74	90.23	90.29	91	N/A
	Threshold based automation	88.58	88.16	88.25	90.25	N/A

randomized occupancy data points, and its predicted appliance adjustments ($\hat{y}$) are compared against ground truth values (y_true). If the AI's assigned setpoint deviates from the predefined range, the prediction is considered incorrect. The accuracy of the system is assessed using precision, recall, and F1-score. The confusion matrix for occupancy-based autonomous building operation with the AI agent for different scenarios is shown in Figure 6.

In addition, to evaluate the AI agent's performance in threshold-based control mode, the study implements predefined environmental control thresholds, including temperature (20°C-27°C), humidity (40%-100%), light intensity (50-150 lux), and carbon monoxide levels (0-50 ppm). These simplified threshold values are implemented primarily to demonstrate the functionality of the LLM-based automation system and its response mechanisms. This experiment involves feeding 400 environmental data points via REST API, where the AI agent continuously monitors real-time temperature, humidity, light intensity, and carbon monoxide levels. If the real-time data falls below or exceeds the predefined threshold, the AI will adjust the corresponding smart appliance by increasing or decreasing its performance to restore the environmental conditions to the comfort range. The AI's adjustments are then validated against predefined expected actions, which are whether to increase or decrease the performance of smart appliances. The confusion matrix for threshold-based autonomous building operation with the AI agent for different scenarios is shown in Figure 7. Accuracy is assessed using precision, recall, F1-score, and macro-averaged performance across all parameters to evaluate the AI's effectiveness in maintaining optimal building conditions.

7 Result and Discussion

The evaluation results in Table 1 show the performance of the proposed virtual assistants and AI agents across different metrics. The virtual assistant for smart appliance control performed exceptionally well, achieving a precision of 95.91%, a recall of 92.75%, and an F1 score of 94.17%, demonstrating high accuracy and reliability in executing user commands. With an accuracy of 92.75% and a reliability score of 95.45%, it proved to be highly reliable in consistent task execution. For the AI agent, the occupancy-based automation system possesses a precision of 90.74%, a recall of 90.23%, and an F1 score of 90.29%, with an accuracy of 91%. These results indicate strong performance in adjusting smart appliances based on occupancy levels. The threshold-based automation system had slightly lower scores, with precision at 88.58%, recall at 88.16%, and an F1 score of 88.25%, achieving an accuracy of 90.25%.

Furthermore, the scalability of the proposed LLM-based AI systems is evaluated based on its throughput and ability to handle concurrent user requests, specifically measuring how many requests the AI can process simultaneously and how quickly the system can respond to user queries. In this experiment, LlamaBench [13], an open-source tool for benchmarking LLM, is used to assess the performance of the proposed AI-based agent and virtual assistant. The results indicated that execution time and throughput varied based on the specific task. For chat or text generation, the average throughput was 33.66 tokens per second. One token is approximately equivalent to 4 English characters, and 1,500 words correspond to around 2048 tokens [14]. Smart home control tasks took longer, with an average execution time of 5402.62 milliseconds per task and throughput of 12.77 tokens per second. Concurrency user request is also an important indicator of the LLM model's scalability [15]. This study used Llamacpp for model deployment, which allows parallelization based on the model context length. For instance, a model with a context length of 8192 tokens can theoretically handle 16 parallel requests where each prompt has 256 tokens, and each response generates 256 tokens [16]. Although the Llama 3 model we used supports a context length of up to 128k tokens, we limited it to 8192 tokens due to limited computational resources. For real-world deployment, especially with a larger user base, we can improve the scalability by opting for models with larger context lengths and running them on machines with greater GPU RAM capacity.

8 Conclusion and Future Works

The proposed LLM-powered virtual assistant in this study allows users to interact with the building through voice and text interfaces for smart appliance control. The AI agent also powers the autonomous building operations by autonomously adjusting smart appliances, such as lighting and HVAC, based on occupancy and the baseline environmental comfort threshold to maintain optimal conditions for occupants. The virtual assistant and AI agent demonstrated strong performance with over 90% accuracy, recall, precision, F1, and reliability. In addition, the current trend towards smaller and more efficient language models, as evidenced by Microsoft's Phi-3, Meta's LLaMA 3.2, and Google's Gemma-2 model, indicates that powerful AI systems will be able to effectively deploy on low-cost edge devices such as Raspberry Pi while offering impressive performance. This development could significantly contribute to extending the reach of smart building systems to entire smart cities, enabling more distributed and responsive urban management.

REFERENCES

[1] S. A. Prieto, E. T. Mengiste, and B. García De Soto, "Investigating the Use of ChatGPT for the Scheduling of Construction Projects," *Buildings*, vol. 13, no. 4, p. 857, Mar. 2023, doi: 10.3390/buildings13040857.

[2] H. You, Y. Ye, T. Zhou, Q. Zhu, and J. Du, "Robot-Enabled Construction Assembly with Automated Sequence Planning Based on ChatGPT: RoboGPT," *Buildings*, vol. 13, no. 7, p. 1772, Jul. 2023, doi: 10.3390/buildings13071772.

[3] H. Chen *et al.*, "Augmented reality, deep learning and vision-language query system for construction worker safety," *Automation in Construction*, vol. 157, p. 105158, Jan. 2024, doi: 10.1016/j.autcon.2023.105158.

[4] S. M. J. Uddin, A. Albert, A. Ovid, and A. Alsharef, "Leveraging ChatGPT to Aid Construction Hazard Recognition and Support Safety Education and Training," *Sustainability*, vol. 15, no. 9, p. 7121, Apr. 2023, doi: 10.3390/su15097121.

[5] J. Zheng and M. Fischer, "Dynamic prompt-based virtual assistant framework for BIM information search," *Automation in Construction*, vol. 155, p. 105067, Nov. 2023, doi: 10.1016/j.autcon.2023.105067.

[6] A. Saka *et al.*, "GPT models in construction industry: Opportunities, limitations, and a use case validation," *Developments in the Built Environment*, vol. 17, p. 100300, Mar. 2024, doi: 10.1016/j.dibe.2023.100300.

[7] Hevner, March, Park, and Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, p. 75, 2004, doi: 10.2307/25148625.

[8] Meta, "Meta Llama 3," Meta Llama. Accessed: May 22, 2024. [Online]. Available: https://llama.meta.com/llama3/

[9] G. Gerganov, *ggerganov/llama.cpp*. (Jul. 04, 2024). C++. Accessed: Jul. 04, 2024. [Online]. Available: https://github.com/ggerganov/llama.cpp

[10] Y. Zhao *et al.*, "Atom: Low-bit quantization for efficient and accurate llm serving," *Proceedings of Machine Learning and Systems*, vol. 6, pp. 196–209, 2024.

[11] "reachsak/LLM_AI_Assistant_for_Human_Building_Interaction." Accessed: Feb. 11, 2025. [Online]. Available: https://github.com/reachsak/LLM_AI_Assistant_for_Human_Building_Interaction

[12] R. Ly, *reachsak/LLM-AI-agent-for-Autonomous-Building-operation*. (Jan. 19, 2025). JavaScript. Accessed: Feb. 11, 2025. [Online]. Available: https://github.com/reachsak/LLM-AI-agent-for-Autonomous-Building-operation

[13] "llamabench," GitHub. Accessed: Oct. 22, 2024. [Online]. Available: https://github.com/ggerganov/llama.cpp/blob/master/examples/llama-bench/README.md

[14] OpenAI, "What are tokens and how to count them?" Accessed: Sep. 23, 2024. [Online]. Available: https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them

[15] Y. Yao *et al.*, "ScaleLLM: A Resource-Frugal LLM Serving Framework by Optimizing End-to-End Efficiency," Sep. 10, 2024, *arXiv*: arXiv:2408.00008. Accessed: Sep. 27, 2024. [Online]. Available: http://arxiv.org/abs/2408.00008

[16] Y. Fu, "LLM Inference Sizing and Performance Guidance," VMware Cloud Foundation (VCF) Blog. Accessed: Sep. 27, 2024. [Online]. Available: https://blogs.vmware.com/cloud-foundation/2024/09/25/llm-inference-sizing-and-performance-guidance/