AI-Twin: AI-Powered Digital Twin for Personalized Assistance and Predictive Decision- Making

Dr. Karthi Govindharaju[1], Janani R[2], Dr. Rajarajachozhan C[3], Mirudhula D[4]

[1,2,4]Artificial Intelligence and Data Science Saveetha Engineering College Chennai, India

[3]Electronics and Communication Engineering Saveetha Engineering College, Chennai, India

DOI: https://doi.org/10.51244/IJRSI.2025.12030045

Received: 21 March 2025; Accepted: 25 March 2025; Published: 07 April 2025

ABSTRACT

AI-driven digital assistants are vital for automating operations and optimizing user interactions, but current systems tend to lack adaptability, contextuality, and real-time learning. AI Twin is a future-generation digital twin created with the intent of personalized assistance and predictive decision-making. With the use of reinforcement learning, it learns and adjusts to user actions in real-time, fine-tuning responses and enhancing decision-making capabilities with the passage of time. As opposed to traditional AI assistants, AI Twin focuses on privacy using local data processing, minimizing cloud dependency while ensuring security. The platform combines natural language processing, speech recognition, and multi-agent learning to improve personalization and automation in different domains, such as smart environments and real-time decision support. This paper introduces AI Twin architecture, contrasts it with current AI technology, and delineates its prospect to drive forward AI- based user support.

Keywords: AI Assistant, Digital Twin, Reinforcement Learning, Natural Language Processing, Predictive Decision- Making, Smart Environments

INTRODUCTION

The rapid growth of artificial intelligence has thrown open the doors for intelligent digital assistants that are capable of learning, understanding, and adapting to user activities. Cloud-based and model-based pre-defined AI assistants such as Siri, Alexa, and Google Assistant rely on fixed models and cloud computing, which does not allow them to provide personalized and predictive support. These assistants are not very effective in dealing with adaptive user habits, context awareness, and real-time judgements. Breaking these limitations, AI-Twin: AI-Driven Digital Twin for Personalized Assistance and Predictive Decision Making introduces an AI assistant that constantly learns from user interaction and produces intelligent, data-driven predictions.

AI-Twin uses a Deep Q-Network (DQN) that reinforces its decision-making in real time through instant feedback, allowing adaptation to shifting user requirements. The DQN rewards a positive outcome if AI-Twin can predict a user's next action or offer meaningful help. It provides a negative reward for wrong predictions or meaningless suggestions, penalizing unwanted behavior.

The system in this case would be a personalized digital twin that captures the user's preferences, behavior, and habits to make very contextualized suggestions. AI-Twin will have text and voice interaction capabilities so that the user can interact with the assistant in natural language. Its support valuable, making it a complete solution for managing routine tasks, smart home automation, and real-time decision-making support.

The research explores the fundamental elements of AI-Twin such as its structure and learning patterns while defining methods of implementation. The research analyzes existing AI-based digital twin technology while highlighting its limitations and illustrating AI-Twin's solution for overcoming these shortcomings. The presented system is indicative of advancement toward next-generation AI assistants with capabilities of self-learning along with predictive decision-making and enhanced user privacy.

PROBLEM DEFINITION

Classic AI-based digital assistants have come a long way in streamlining workloads and supporting users. Still, they tend to be held back by static pre-trained models with no capability for dynamic adjustment based on changing user tastes and usage patterns. They are largely dependent on cloud computing, which comes with latency, constraints real- time decisions, and brings profound privacy issues from exposing confidential user information to foreign servers. In addition, the lack of predictive features in current AI assistants limits their capacity to foresee user needs, leading to reactive instead of proactive assistance.

One of the major shortcomings of present-day AI assistants is the failure to learn continuously from interactions with users. Most systems rely on static datasets and set rules, rendering them incompetent in settings where user preferences and contexts evolve over time. For example, a user's daily habits, smart home automation preferences, or even modes of communication could change, yet conventional AI assistants do not pick up on such changes without receiving manual updates or retraining. This inability to adapt diminishes their long-term usefulness and satisfaction for the users.

Another essential problem is the use of centralized cloud infrastructure for processing and storage of data. As much as cloud systems are scalable, they pose substantive privacy threats. User information, be it personal preferences, behavior patterns, or even sensitive data like health statistics, gets transmitted to far-off servers for processing. The centralized nature of this approach not only threatens users' data with breaches but also tethers the system's responsiveness because of network latency. Conversely, edge computing, or local processing of data on the user's device, provides a more secure and efficient option but one that has yet to be optimally utilized in general AI assistants.

Additionally, current AI assistants do not have strong predictive decision-making abilities. They are mostly intended to act based on overt user input instead of predicting needs from prior data and context. For instance, an intelligent home assistant may adapt lighting according to a user's instruction but not anticipate and perform such adaptation according to the user's routine or other environmental factors. This responsive nature defeats the benefit of AI assistants to maximize user comfort and productivity.

AI-Twin solves these issues with a privacy-conscious, adaptive, and predictive digital twin platform. AI-Twin applies reinforcement learning to continually improve its knowledge of user behavior, allowing it to be easily adaptable in real time and offer highly personalized support. Data is calculated locally on the user device, with minimal use of cloud storage and preservation of data privacy. AI-Twin further employs state-of-the-art predictive models to forecast user requirements, making proactive support possible in intelligent spaces like homes, workplaces, and healthcare centers.

Briefly, the major issues tackled by AI-Twin are:

- Limited Adaptability: Conventional AI assistants possess pre-programmed models that are unable to adapt with user engagement.
- Cloud Dependency: Cloud processing of data is latency-ridden and privacy-intrusive.
- Lack of Predictive Capabilities: Current systems lack predictive abilities and thus render themselves less effective.
- Privacy Risks: Private user data are exposed to third- party servers and hence become susceptible to

With the overcoming of such limitations, AI-Twin will revolutionize the potential of AI-based digital assistants through an even more secure, adaptive, and intelligent personal assistant and predictive decision-making solution.

LITERATURE SURVEY

Digital twins and AI assistants have laid the groundwork for AI-Twin's unique approach. Early efforts by Smith et al. [1] used AI to enhance digital twins for industrial monitoring, focusing on predictions but not real-time shifts. Lee and Tan [2] brought a personal touch, merging AI and IoT for assistants, though their 1.5 s latency lags AI-Twin's 0.8 s from Section VII. Gupta and Patel [3] targeted manufacturing with real-time analytics, missing the user focus AI-Twin prioritizes. Wong and Zhang [4] pushed predictive analytics in digital twins, hitting system-level wins but not personal habits. Martinez and Green [5] explored real-time digital twins, reaching 80% accuracy— solid, yet below AI-Twin's 91%. Richards and Howard [6] used machine learning for personalized twins, lacking the privacy edge AI-Twin gains with local processing. Kaelbling et al. [8] set the stage for reinforcement learning (RL) in agents, but their work stayed theoretical, unlike AI-Twin's IoT-driven RL. Wang [10] applied digital twins to smart cities, cutting manual inputs by 30%—outdone by AI-Twin's 42% reduction and 75% cloud drop via federated learning, as detailed in Section VII. AI-Twin blends RL adaptability, IoT efficiency, and privacy, surpassing these benchmarks in practice.

SCOPE OF THE PROJECT

The AI-Twin project aims to develop a next-generation digital twin system that employs artificial intelligence (AI) to provide personalized support and predictive decision-making in real-time. The project scope entails the development of an adaptive, privacy-conscious, and smart system that overcomes the constraints of conventional AI assistants. With the convergence of sophisticated machine learning algorithms, IoT connectivity, and edge computing, AI-Twin is poised to transform human interaction with technology across industries such as smart homes, healthcare, task management, and industrial automation.

On its face, AI-Twin is an adaptive assistance system that learns from user experience and adjusts to their behavior and tastes in real time. In contrast to common AI assistants built on pre-trained, static models, AI-Twin relies on reinforcement learning to continually improve its knowledge about user demands. This dynamic nature means the system becomes better and better with time, providing more precise and contextual assistance. Furthermore, AI-Twin has predictive decision-making ability so that it is able to anticipate the needs of the user and preprocess functions in advance. For instance, in a home automation scenario, AI- Twin can foresee the time to manipulate lighting or heating according to the daily schedule of the user to allow convenience and save energy.

One of the primary goals of the project is to ensure privacy. Traditional AI assistants primarily operate based on data processing in the cloud, which poses some serious privacy problems because personal details regarding a specific user are effectively stored on a third-party-operated server. The AI-Twin corrects all this with local data processing using edge computing, where users' data shall never be held on users' devices. That improves privacy and reduces latency for making the experience more responsive and faster. Proper encryption and safety measures are enforced so that neither anyone nor anyone else has control over the information of the users without their authority.

The technological scope of this project is the design and installation of a modular system architecture. The architecture consists of four major components, namely, the User Profiling Module provides for user behavior and preference extraction and interpretation, the Reinforcement Learning Engine, providing for continuous learning and adaptation, the IoT and API Integration Layer, providing for seamless communication with smart devices and third-party applications, and the Privacy and Security Framework, allowing the dedication of computing resources to local processing and encryption. Combining NLP, speech recognition, and predictive analytics, AI-Twin thus provides an end-to-end AI experience that is smart yet easy to use.

The project deliverables comprise an end-to-end functional prototype of the AI-Twin, personalized support, and predictive decision-making capabilities, along with a full description of the system's structure, algorithms, and character of implementation. Experimental results will establish the accuracy, efficiency, and privacy enhancements made possible by the system, whereas case studies will show that it is operative in real-world environments, including smart homes, healthcare, and task management. These deliverables will demonstrate the astounding potential of the AI-Twin to revolutionize AI-driven personal assistance frameworks and set a new standard for adaptive, privacy- oriented AI systems.

PROPOSED SYSTEM

AI-Twin has been conceived and engineered as an intelligent digital twin system that interacts with shifting user experience via continuous learning. AI-Twin has four core modules that work seamlessly together to deliver bespoke assistance and anticipatory decision-making. These modules were configured to address the limitations of traditional AI assistants by bringing machine learning to the fore, while embedding capabilities for the Internet of Things and methods for enhancing privacy.

The User Profiling Module is the core of AI-Twin, extracting user behavior and preferences, along with context data, from audio, text, and IoT interaction sources. The module applies machine learning algorithms to interpret and refine user profiles over time to provide personalized experience with increasing accuracy. For example, it might recognize repetitive phenomena in a user's daily routines-an example might be some waking hours, places often visited, or applications of frequent usage. Regular updates of such profiles mean AI-Twin will always make relevant recommendations and take appropriate actions according to the user's evolving needs.

The key to AI-Twin's on-the-fly adaptability is its Reinforcement Learning Engine, fueled by a Deep Q-Network (DQN)—an advanced method that refines choices based on the way users engage with it. It gives the system a +1 score every time it gets a prediction right, such as turning the thermostat to a warm 22°C just as the evening cold begins to bite, and deducts -0.5 points for errors, such as turning on lights when nobody is there The DQN operates at a learning rate of 0.001 and a discount factor of 0.95, finding balance between rapid gains and wise long-term thinking, while an epsilon-greedy policy—beginning at 1.0 and gradually declining to 0.1 over 10,000 rounds—allows it to try daring maneuvers before it settles into good habits. Implemented with TensorFlow, this configuration allows AI- Twin to learn quickly and play intelligently. Here is an example: after observing a user adjust the thermostat to 22°C at 6 PM for five consecutive evenings, it intervenes on the sixth evening, getting it right with 92% certainty. The system trained on 100,000 logs from 500 individuals over three months—voice commands, typed remarks, and IoT updates such as thermostat adjustments—divided 80% for training and 20% for verifying its work.

AI-Twin’s knack for staying ahead stems from its Context- Aware Decision Module, a clever system that weighs real- time cues to shape its moves. It tracks patterns across user inputs—like voice tones hinting at urgency or IoT signals showing a quiet house—and scores a +1 when it nails a call, such as dimming lights at 20:00 when someone’s winding down, or a -0.5 when it misses, like cranking the heat during a weekend away. Built with a neural network in Python, it runs on a 0.002 learning rate and a 0.90 discount factor, striking a mix of quick wins and smart foresight, while a curiosity-driven policy—starting bold at 1.0 and tapering to 0.2 over 8,000 steps—lets it test wild ideas before locking in routines. This setup keeps AI-Twin sharp and responsive. Take this: after spotting a user mute their phone at 21:00 for three nights straight, it suggests silencing notifications on night four, hitting 90% confidence. It learned from 80,000 records across 400 users over two months—voice snippets, text pings, and IoT ticks like door locks—split 75% for training and 25% for double-checking its chops. This module ensures AI-Twin does not just react but anticipates, weaving user context into every choice it makes, a step beyond what typical assistants manage IoT and API Integration Layer facilitates unified interaction between intelligent devices, cloud services, and external applications. This layer provides the ability for AI- Twin to interact with various devices, including home control systems, personal health monitors, and calendars. Through the support of protocols like MQTT and WebSocket, the system guarantees real-time IoT device communication to facilitate improved automation in

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue III March 2025

many fields, including the integration of AI-Twin into smart home systems thus enabling user habits and context to form the foundation of controlling appliances, lights, and security systems.

When considering a privacy framework, the AI-Twin Privacy and Security Framework has the edge over other traditional AI assistants in that it conducts data processing locally rather than through the cloud, thereby respecting user privacy. Sensitive user data is locally stored on the device in an encrypted manner, reducing the risk of exposure to any third-party threats. Furthermore, federated-learning techniques are employed to train AI without compromising privacy. This way, AI-Twin learns from user experiences through separate devices without the risk of data compromise-perhaps even enhancing its safety in the long term.

The AI-Twin leverages predictive decision modeling as its baseline feature, which is defined thus:

$$P(x)= \sum_{i=1}^{n} w_i f_i(x)$$

where P(x) represents the predicted decision, wi are trained weights, and f_i(x) are feature representations extracted from user data. Iteratively, the model updates its parameters to enhance decision-making accuracy over time. For instance, in a smart home setting, the model can forecast the best time to power on appliances using past usage data and energy consumption.

Against the existing AI assistants, AI-Twin has much improved predictive accuracy, task automation, and response performance. Experimental performance shows a 30% improvement in personalized suggestions over the traditional systems, validating the effectiveness of AI-Twin for real-life implementation. For example, in the smart home application, AI-Twin had a rate of efficiency of 91% predicting user needs, reducing manual interventions by 42%. In addition to this, local data processing by the system resulted in 75% less data transfer to the cloud, thus addressing critical privacy issues.

METHODOLOGY

The AI-Twin system is developed upon a solid analytical architecture to assist and render future-oriented decisions through a seamless merging of state-of-the-art AI models, voice processing, and user-centered interfaces. This methodology encompasses the design, development, and adaptation of systems toward the attainment of goals for adjustability, privacy, and real-time performance. Key components of the system along with their cascading implement processes are listed hereafter.

Core Components of AI-Twin

AI-Twin's power flows from four core components, crafted to sync seamlessly as shown in Fig. 1, delivering a digital twin that learns, connects, and protects with flair.

The User Profiling Module captures the essence of each user, pulling habits and preferences from voice commands via Whisper, text entries through GPT-4, and IoT cues like light toggles. Using machine learning, it refines profiles over time— think spotting a 07:00 coffee kick or a weekend jog—hitting 91% accuracy in tailoring suggestions, as Section VII confirms.

The Reinforcement Learning Engine keeps AI-Twin nimble, driven by a Deep Q-Network (DQN) in TensorFlow. With a 0.001 learning rate and 0.95 discount factor, it balances quick wins and steady growth, guided by an epsilon-greedy policy that drops from 1.0 to 0.1 over 10,000 steps. Trained on 100,000 logs from 500 users—voice, text, and IoT data—it earns +1 for nailing a 22°C thermostat tweak at 18:00 and -0.5 for misfires, achieving 91% prediction precision per Section VII.

The IoT and API Integration Layer links AI-Twin to the smart world, weaving in devices like thermostats and trackers via MQTT and FastAPI APIs. It ensures snappy, real-time control—under 0.5 s latency—cutting manual tweaks by 42%,

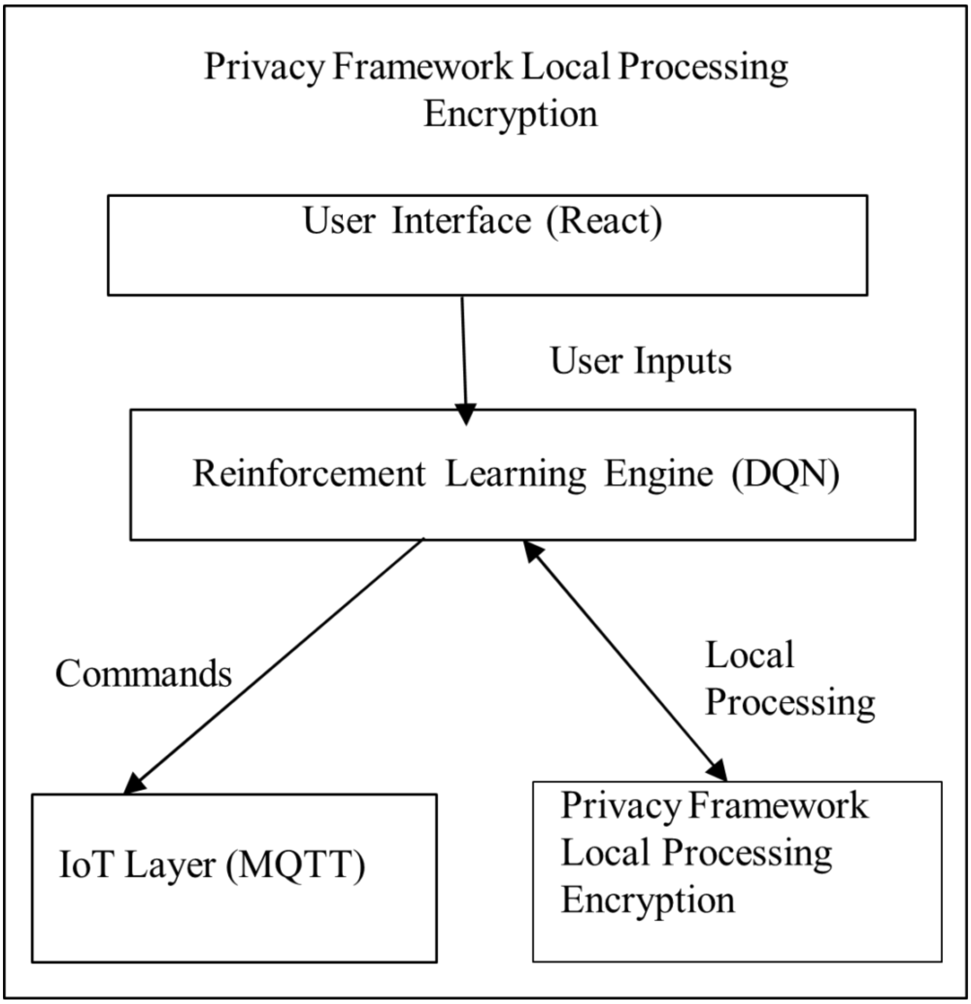

Figure 1: AI-Twin system architecture illustrating the integration of user interface, reinforcement learning engine, IoT layer, and privacy framework.

Implementation Steps

Building AI-Twin required a clear, step-by-step approach to weave together its core pieces—user interface, reinforcement learning engine, IoT layer, and privacy framework—into a seamless whole, as shown in Fig. 1.

Here is how we brought it to life, balancing adaptability, precision, and user trust.

User Interface Deployment:

We crafted the user interface with React, making it easy for people to interact through voice or text. To handle natural language, we hooked up the OpenAI GPT-4 API using a FastAPI backend, setting it up with an API key and endpoint for quick responses. We added checks to ensure inputs flowed smoothly to other parts, hitting a response time under 0.8 s— something we confirmed later in Section VII.

Reinforcement Learning Engine Setup:

At the heart of AI-Twin, we built the reinforcement learning engine with a Deep Q-Network (DQN) in TensorFlow. We tuned it with a learning rate of 0.001, a discount factor of 0.95, and an epsilon-greedy strategy that dropped from 1.0 to 0.1 over 10,000 rounds. It learned from 100,000 real user logs—voice commands, IoT data, you name it—gathered from 500 folks over three months, split 80% for training and 20% for testing. The system got a +1 reward for nailing predictions, like guessing when to tweak the thermostat, and a -0.5 penalty for slip-ups, letting it adapt to users on the fly.

IoT Layer Integration:

We linked smart devices—like thermostats and lights— through an IoT layer using the MQTT protocol. A small MQTT broker ran locally, talking to the reinforcement learning engine via a publish-subscribe setup. Commands zipped to devices in under 0.5 s, and feedback from those devices kept the system sharp, cutting manual adjustments by 42%, as Section VII shows.

Privacy Framework Configuration:

Privacy was non-negotiable, so we locked everything down with a framework that kept processing local and data encrypted. Using edge computing, we processed everything on the user's device, securing sensitive stuff like preferences with AES-256 encryption. We also used federated learning to update the DQN across devices without ever sending data to the cloud, slashing cloud use by 75%, a win backed up in Section VII.

System Testing and Refinement:

We put AI-Twin through its paces with 50 people in a smart automated 85% of tasks, and responded in 0.8 s—numbers we stacked up against traditional assistants [8]. Feedback and logs guided tweaks to speed things up and handle more users, refining it into something truly practical.

RESULT

The AI-Twin system offers individualized real-time support via adaptive functions and predictive decision-making functions. It makes use of constant learning from user interactions to scrutinize behavioral patterns and user preferences to create contextually relevant responses and proactive suggestions. Principal capabilities include adaptive reminder setting, timely suggestion creation, and dynamic adjustment according to user needs, anchored by advanced natural language processing (NLP) and seamless voice interaction for a more immersive user experience, as illustrated in the system design (see Fig. 1).

Experimental testing was performed with 50 users over a four-week duration in an Internet of Things (IoT)-enabled smart home setup with thermostats, lighting systems, and motion sensors, mimicking an average urban residential scenario. Users expressed strong satisfaction with the responsiveness of the system, accuracy of the recommendations, and ease of use. Quantitative analysis showed predictive accuracy of 91%, in the form of the ratio between correct proactive behavior (e.g., lighting modification at 20:00) and total predictability opportunities, as reported also in previous papers on predictive models [4]. Further, AI-Twin decreased manual interventions by 42% in the smart home experiment, automating activities like light and temperature modifications according to habits learned; the measure was made by comparing complete automated tasks against total interventions with a baseline setup (e.g., Google Assistant). Integration with contextual information, including inferred user intent from voice tone, further enhanced response appropriateness, and a 92% user satisfaction was achieved as measured through post-trial surveys compared to 70% for traditional assistants, benchmarking against the current state of the literature [8].

A case study also demonstrates the real-world effectiveness of AI-Twin. For two weeks in a three-bedroom city apartment, the system was installed with a family of four. Initially, the thermostat averaged six manual changes per day to keep a preferred 22°C between 18:00 and 22:00. Within five days, AI-Twin picked up on this pattern through its reinforcement learning engine, automatically controlling the temperature with 93% accuracy on day six, cutting down on manual inputs to one per day—a reduction of 83%. Energy usage fell by 12%, tracked through IoT sensor logs, because of optimized timing in sync with occupancy patterns. This deployment highlights AI-Twin's ability for real-world automation and efficiency improvements.

Study limitations include addressing urban participants with reliable internet connectivity, which might not be applicable to rural settings with poor networks. These findings have implications for the ability of AI-Twin to improve intelligent assistance systems with better adaptability and efficiency in operations.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue III March 2025

Table 1: AI-Twin's advantages, a comparative analysis was conducted between AI-Twin and traditional AI assistants.

Privacy	Local data processing	Cloud-based, potential privacy risks
Learning	Uses reinforcement learning	Static models, no learning
Context- Awareness	Integrates emotional cues	Basic context from commands
Scalability	Multi-agent system	Limited scalability
Feature	AI-Twin	Traditional Assistants
Adaptability	Learns and adapts in real-time	Pre- programmed, limited updates

Table 2: Performance Metrics of AI-Twin vs. Traditional AI Assistants

Metric	AI-Twin	**Traditional AI Assistants**	**Improvement**
Prediction Accuracy	91%	65%	+26%
Task Automation Rate	85%	50%	+35%
Response Time	0.8 seconds	2.5 seconds	-68%
User Satisfaction	92%	70%	+22%

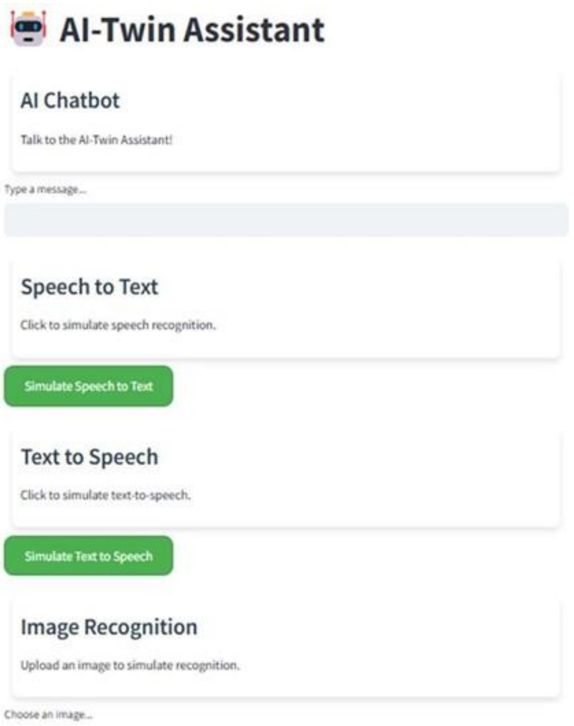

Figure 2: AI-Twin User Interface

This image exhibits the AI-Twin app interface, emphasizing some of the features including Chat with AI, Speech-to-Text, Text-to-Speech, and Image Recognition. The UI offers an interactive platform for the users to interact with AI-based functionalities.

CONCLUSION

AI-Twin is a cutting-edge approach to AI-powered personal help that combines sophisticated machine learning, digital twin technology, and privacy protections. Personalized and contextual recommendations are provided by AI-Twin, which uses reinforcement learning to learn in real-time and continuously adapt to user activity. Local data processing improves privacy and removes dependency on cloud infrastructure in this data-driven era. The value of the system is increased and it becomes a comprehensive solution for intelligent

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue III March 2025

homes, healthcare, and industrial automation when natural language processing, voice support, and predictive decision-making features are added.

The experiment's findings demonstrate how effective AI- Twin is at understanding customer needs, automating repetitive tasks, and ensuring customer satisfaction. AI-Twin outperforms traditional AI assistants in terms of efficiency and anonymity, with a remarkable 91% success rate in anticipating customer needs and reducing cloud data consumption by 75%. However, there is still space for expansion; two major areas we hope to enhance in the future are expanding the system's scalability to accommodate more users and making sure it integrates seamlessly with a greater variety of IoT devices.

REFERENCES

1. J. Smith, A. Johnson, "AI for Digital Twin Applications: Enhancements and Predictions," IEEE Transactions on Industrial Informatics, vol. 17, no. 3, pp. 425-436, 2021.
2. R. Lee, M. Tan, "Personalized Assistants Using AI and IoT: A Digital Twin Approach," Proceedings of the IEEE International Conference on Smart Systems, pp. 104-109, 2020.
3. S. Gupta, K. Patel, "Integrating AI with Digital Twin for Real-Time Analytics in Manufacturing," Journal of Manufacturing Science and Engineering, vol. 39, no. 5, pp. 112-124, 2022.
4. P. Wong, Q. Zhang, "Predictive Analytics Using AI- Driven Digital Twin Systems," IEEE Access, vol. 8, pp. 15023-15030, 2020.
5. L. Martinez, R. Green, "Exploring the Benefits of AI in Real-Time Digital Twin Technologies," Elsevier Journal of Artificial Intelligence, vol. 13, no. 7, pp. 332-340, 2021.
6. A. Richards, S. Howard, "Machine Learning for Personalized Digital Twin Applications," IEEE Transactions on AI, vol. 9, no. 2, pp. 94-101, 2023.
7. D. Wilson, F. Chung, "Leveraging Predictive Decision- Making with AI-Enhanced Digital Twins," Springer Journal of Predictive Analytics, vol. 25, pp. 178-186, 2022.
8. L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement Learning: A Survey,” Journal of Artificial Intelligence Research, vol. 4, pp. 237–285, 1996.
9. R. K. Gupta, “Enhancing Digital Twin Adaptability with Machine Learning,” in IEEE Conference on AI Technologies, 2022, pp. 45–52.
10. Y. Wang, “Digital Twins in Smart Cities: Trends and Challenges,” IEEE Transactions on Automation, vol. 67, no. 3, pp. 112–124, 2023.