



Journal of Interdisciplinary Science & Technology

Toward Autonomous and Self-Evolving E-Commerce: A Review of Multi-Agent Personalization, Customer Digital Twins, Constrained Generative Interfaces, and Safe Policy Learning

Devendra Singh Kushwaha¹, Dr. Ankita V. Karale²,
Rushikesh Khule³, Ankita Bhoi³, Pooja Pakhale³, Punam Ahire³

¹Assistant Professor, Department of Computer Engineering, Sandip Institute of Technology and Research Center, Nashik-422213, India

²Head, Department of Computer Engineering, Sandip Institute of Technology and Research Center, Nashik-422213, India

³Department of Computer Engineering, Sandip Institute of Technology and Research Center, Nashik-422213, India

devendra2904@gmail.com, ankita.karale@sitrc.org, rushikhule5@gmail.com, ankubhoi.777@gmail.com,
poojap112004@gmail.com, psahire2004@gmail.com

Corresponding author- Devendra Singh Kushwaha

DOI: 10.5281/zenodo.23533726

Abstract: Personalization in electronic commerce is commonly implemented as a collection of separately optimized services for recommendation, search, advertising, and content selection. Such systems can identify relevant products while still presenting them through an unsuitable interface, an inappropriate explanation, or an action that conflicts with accessibility, privacy, inventory, or business constraints. This review examines the convergence of four research directions: behaviour-aware customer modelling, large-language-model-based multi-agent systems, constrained generative user interfaces, and reinforcement-learning-based policy improvement. On the basis of this synthesis, the paper proposes a reference architecture for an autonomous, self-evolving e-commerce platform. The architecture maintains a purpose-limited customer digital twin, infers session intent with calibrated uncertainty, collects typed proposals from specialist agents, and produces a declarative page specification over an approved component library. A deterministic policy layer validates every proposed adaptation before rendering. Behavioural and explicit preference signals are combined through recency-, relevance-, and confidence-aware scoring rather than a fixed event hierarchy. Learning is deliberately offline-first: candidate policies are evaluated from logged or simulated trajectories and promoted only after safety, accessibility, latency, and stability checks. The paper further defines research questions, baselines, metrics, and an ablation plan. The resulting framework positions generative AI as a bounded decision-support mechanism rather than an unrestricted source-code generator, thereby preserving auditability and operational control.

Keywords: e-commerce personalization, multi-agent systems, generative user interfaces, digital twins, safe policy learning, reinforcement learning.

I INTRODUCTION

Electronic commerce has progressed from catalogue retrieval to a decision-intensive environment in which presentation, explanations, and timing heavily influence customer behaviour. Modern recommender systems can learn expressive product representations from large interaction logs [19], [20], [22]. However, most deployment architectures treat layout design, search ranking, promotion delivery, and content generation as isolated problems. This separation creates friction: a recommendation engine may identify an item with high predicted utility, but the surrounding layout may hide its key attributes, emphasize irrelevant urgency cues, or violate accessibility guidelines.

Several developments create an opportunity to address this limitation. Generative agents powered by large language models (LLMs) can perform multi-step reasoning, tool execution, and collaborative task decomposition [1], [2], [3], [5], [6]. Concurrently, generative user interface (UI) frameworks enable software to synthesize declarative layout specifications dynamically rather than serving static templates [15], [16]. When combined with customer digital twins—data-connected models that maintain behavioral state, preferences, and privacy boundaries [25]—it becomes possible to personalize both what is presented and how it is presented.

Yet, deploying generative agents in high-stakes retail environments introduces significant risks. Unconstrained LLM

outputs may hallucinate non-existent product features, violate price constraints, produce erratic UI shifts that confuse users, or leak sensitive behavioural inferences. Furthermore, updating online policies based directly on real-time customer interactions can destabilize user experience or expose the platform to adversarial exploitation. Safe operational deployment requires strict policy guardrails, deterministic execution limits, and an offline-first learning cycle [23], [24].

This paper is a structured narrative review and design synthesis rather than a report of completed empirical benchmarks. It makes five contributions:

- It reviews the relationship among recommender systems, customer-state modelling, multi-agent frameworks, generative interfaces, and policy optimization.
- It proposes a modular reference architecture in which specialist agents emit typed proposals that are validated by a deterministic policy orchestrator before rendering.
- It defines a signal-prioritization method that combines purchases, cart activity, searches, and clicks using recency, relevance, and confidence factors.
- It presents an offline-first evaluation protocol with static, rule-based, single-agent, and bounded multi-agent conditions.
- It identifies technical and ethical risks and maps them to concrete governance controls.

II. REVIEW METHOD AND SCOPE

The source literature was organized around five themes: (1) language-aware recommendation, (2) generative agents and multi-agent coordination, (3) adaptive interfaces and generative UI, (4) behavioural simulation and agent evaluation, and (5) policy optimization. The primary literature search selected peer-reviewed conference and journal publications together with prominent preprints from 2022 to 2024.

The review is narrative and architecture-oriented. It does not claim the coverage of an exhaustive systematic literature review; instead, it synthesizes design principles to answer three primary questions:

- What customer state is required to coordinate recommendation and interface adaptation?
- Which decisions benefit from role-specialized agents, and which must remain deterministic?
- How can a platform improve from behavioural outcomes without allowing uncontrolled online experimentation?

The proposed architecture is consequently a research hypothesis and implementation blueprint. Performance claims must be established by the experimental protocol in Section VII.

III. BACKGROUND AND RELATED WORK

A. Language-Aware Recommendation

Traditional recommender systems estimate affinity from interactions, collaborative filtering, or dense feature embeddings. Modern language-aware recommenders reframe recommendation as text generation or sequence prediction. TALLRec demonstrates that lightweight tuning can align LLMs with recommendation tasks [20]. Broader surveys describe how LLMs may act as feature encoders, scoring functions, or conversational dialogue managers [22]. However, language models evaluated solely on item ranking often lack operational awareness of UI constraints, inventory availability, and business rules.

B. Generative Agents and Multi-Agent Coordination

Generative Agents demonstrates how LLM-powered entities can maintain memories, form reflections, and execute complex social behaviors [1]. Multi-agent systems allocate responsibilities across specialized roles. AutoGen [3], MetaGPT [5], and ChatDev [6] show that role separation reduces prompt complexity and improves task success. Refine mechanisms such as Reflexion [7] and Voyager [8] enable agents to optimize strategies from feedback.

Directly transferring open-ended agent conversation to e-commerce is risky. Agents operating without strict schema constraints can produce non-deterministic layout specifications or conflicting business logic. Effective coordination requires typed proposal channels and deterministic conflict resolution.

C. LLMs and Adaptive User Interfaces

LLMs have been studied as an interaction layer for graphical and conversational interfaces. Wang et al. demonstrate that LLMs can translate natural language into structured UI component commands [15]. LUGI and related systems show that generative models can adapt screen density and component hierarchy [16].

There is, however, an important distinction between generating source code and generating an interface specification. Synthesizing executable code at runtime creates security and stability vulnerabilities. A safe generative UI system must produce declarative page specifications over a pre-approved, accessible component registry.

D. Digital Twins and Behavioural State

A digital twin is a data-connected representation maintained for a defined purpose [25]. Although much digital twin literature focuses on physical systems, customer digital twins represent user intent, historical preferences, device context, and governance boundaries.

The state must distinguish observation from inference. “Item added to cart” is an observed event; “high price sensitivity” is

an inference with associated uncertainty. A reliable digital twin must maintain confidence estimates, temporal decay, and explicit consent boundaries.

E. Learning from Behavioural Outcomes

E-commerce adaptation is sequential: an early layout can alter later searches, comparisons, and purchases. Reinforcement learning (RL) offers tools for multi-step policy optimization [23].

Nevertheless, unconstrained online RL is inappropriate for early deployment. Real traffic may experience layout instability or poor recommendations. Offline policy optimization (OPO) and reinforcement learning from human/rule feedback (RLHF) allow candidate policies to be evaluated safely on logged interaction trajectories prior to deployment [24, 32].

IV. PROBLEM FORMULATION

Let a session for customer u at time t contain consented observations $O_{u,1:t}$, explicit preferences E_u , device context $D_{u,t}$, and market conditions $M_{u,t}$. The platform maintains a customer state:

$$S_{\{u,t\}} = f_{\theta}(O_{\{u,1:t\}}, E_u, D_{\{u,t\}}, M_{\{u,t\}}) \quad (1)$$

where f_{θ} combines deterministic feature aggregation with a sequential state encoder. An intent module estimates a probability distribution over discrete session goals:

$$p_{\{u,t\}} = P(z_{\{u,t\}} | S_{\{u,t\}}), z_{\{u,t\}} \in \{\text{explore, compare, purchase, leave}\} \quad (2)$$

At a decision point, N specialist agents produce candidate adaptation proposals:

$$q_i = (a_i, \hat{v}_i, c_i, E_i, K_i, \tau_i) \quad (3)$$

where a_i is a candidate action, $\hat{v}_i$ is estimated utility, c_i is confidence, E_i is evidence, K_i is constraints, and τ_i is proposal expiry. The orchestrator selects an action bundle A^* that maximizes expected utility under hard guardrails:

$$A^* = \arg \max \hat{U}(A, S_{\{u,t\}}) - \lambda C(A), \text{ s.t. } g_j(A, S_{\{u,t\}}) \leq 0, \forall j \quad (4)$$

where $C(A)$ represents instability or latency costs, and g_j encodes hard privacy, accessibility, and business constraints.

V. PROPOSED REFERENCE ARCHITECTURE

The proposed reference architecture organizes e-commerce personalization into six modular functional layers supported by a central data and state management core. The system decouples creative proposal generation from execution authority.

A. Consent-Aware Event Acquisition

The event collector ingests page views, clicks, queries, dwell intervals, scroll depth, and cart modifications. Consent acts as a functional filter rather than a static banner: users can toggle data sources (e.g., search history vs. purchase history), which instantly updates the data stream available to feature encoders.

B. Customer Digital Twin

The digital twin contains four separable state layers:

- 1) *Observed State*: consented raw interaction logs, timestamps, and device properties.
- 2) *Derived Features*: recency-weighted category affinity, price tier interaction, search frequency.
- 3) *Inferred State*: intent probabilities $p_{\{u,t\}}$, abandonment risk, preference uncertainty.
- 4) *Governance State*: active privacy masks, retention limits, and explicit exclusion flags.

C. Explicit Preference Elicitation

To overcome cold-start limitations, an optional onboarding or settings drawer allows users to specify explicit presentation preferences:

- Visual Character: minimal, product-dense, or story-driven.
- Information Density: compact, balanced, or spacious.
- Content Emphasis: technical specs, customer reviews, high-resolution imagery, or pricing.
- Permitted Data Sources: toggle tracking for session browsing, history, or demographics.

D. Contextual Signal Prioritization

Rather than applying a rigid hierarchy (e.g., purchase > cart > click), the system computes dynamic weights for each behavioural signal:

Table I illustrates contextual interpretation across common shopping scenarios.

Table 1: Contextual interpretation of personalization signals

Signal Pattern	Primary Intent	Risk / Conflict	Recommended Action
Spec comparisons	Comparison	Overwhelming banners	Promote specs matrix
Cart idle dwell	Hesitation	Premature discount	Surface shipping/warranty
High scroll speed	Exploratory	Dense tabular data	Switch to visual grid
Exact query number	Known-Item	Unexpected re-rank	Direct product path

E. Specialist Proposal Modules

The architecture divides responsibilities among five specialist modules:

- UX Specialist: proposes layout families, component ordering, density, and color themes.
- Recommendation Specialist: ranks items, bundles, and complementary accessories.
- Content Specialist: generates grounded, schema-compliant product summaries and feature highlights.
- Offer-Policy Specialist: evaluates eligibility for active promotions, volume discounts, or loyalty perks.
- Analytics Specialist: predicts conversion probability, expected revenue, and latency impact.

F. Decision Orchestration

The decision orchestrator acts as a deterministic gateway. It evaluates incoming proposals q_i through five sequential steps: (1) schema validation, (2) constraint checking (privacy, accessibility, inventory), (3) conflict resolution, (4) utility optimization, and (5) fallback verification. If a proposal fails validation or times out, the system seamlessly defaults to a baseline static layout.

G. Constrained Generative UI Engine

The UI engine translates accepted proposal bundles into a declarative page specification (e.g., JSON UI schema). Components are drawn exclusively from a versioned, pre-tested component library. Executable code generation is strictly prohibited.

H. Model Gateway, Cost, and Caching

To control API overhead and latency, the Model Gateway manages LLM calls via semantic caching, prompt compression, and token budgets. Routine adaptations rely on lightweight statistical rules, reserving full LLM reasoning for high-uncertainty or high-value decision points.

I. Explanation and Auditability

Every adaptation decision logs an audit trace containing the digital twin state version, active policy ID, submitted agent proposals, validation results, and final rendered specification. Users can view plain-language explanations (e.g., “Shown based on your recent spec comparison”).

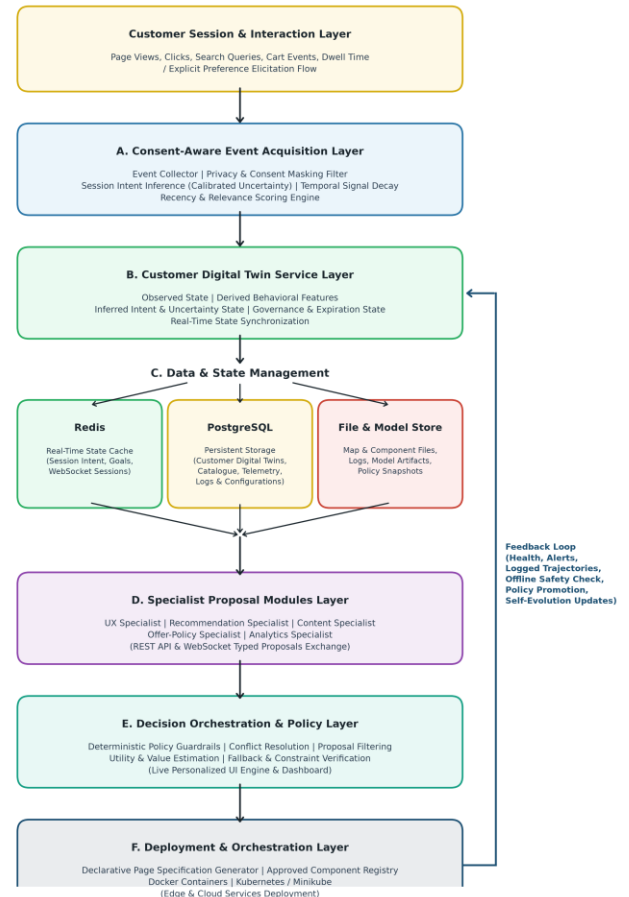


Figure 1: System Architecture Diagram of the Proposed Autonomous and Self-Evolving E-Commerce Personalization Platform.

VI. CONTROLLED SELF-EVOLUTION

The term “self-evolving” refers to guarded, offline policy refinement rather than unconstrained live code modification. The platform evaluates logged interaction trajectories using a multi-objective reward function:

$$R_t = \alpha_1 R_{\text{task}} + \alpha_2 R_{\text{purchase}} - \beta_1 P_{\text{abandon}} - \beta_2 P_{\text{latency}} - \beta_3 P_{\text{instability}} \quad (5)$$

Candidate policies are trained offline, subjected to counterfactual evaluation, and tested against regression suites. Only policies passing safety, accessibility, and stability thresholds are promoted to live canary deployment.

VII. EVALUATION FRAMEWORK

A. Research Questions

- RQ1: Does coordinated UI and recommendation personalization increase task completion compared to static baselines?

- RQ2: Does a bounded multi-agent design outperform single-agent LLM approaches in schema compliance and latency?
- RQ3: Do explicit preference controls reduce cold-start friction and improve user trust?
- RQ4: Does constrained declarative UI generation eliminate invalid or inaccessible layout states?
- RQ5: Can offline policy learning improve cumulative reward without increasing guardrail violations?

B. Baselines and Experimental Conditions

Four experimental conditions are specified for empirical comparison: (1) Static Storefront, (2) Rule-Based Personalization, (3) Single-Agent Generative UI, and (4) Bounded Multi-Agent Reference Architecture.

C. Tasks and Data Splits

Experiments evaluate four user workflows: exploratory browsing, known-item purchase, specification comparison, and cart re-engagement. Evaluation splits rely on temporal cutoffs to prevent data leakage.

D. Metrics

System evaluation spans outcome families detailed in Table II.

Table 2: Proposed evaluation metrics

Metric Family	Primary Indicators	Target
Business Success	Task Completion, Conversion Rate	Maximize
User Trust & UX	Layout Stability, Control Score	Maximize
System Reliability	p95 Latency, Violation Rate	Minimize
Safety & Compliance	Hallucination Rate, Audit Pass %	Zero Violations

E. Ablation and Stress Testing

Ablation studies progressively disable explicit preferences, digital twin memory, or specialist modules. Stress testing evaluates system resilience under malformed agent responses, high network latency, and conflicting user signals.

F. Statistical Reporting

All empirical trials must report sample sizes, confidence intervals, effect sizes, and p-values corrected for multiple hypothesis testing.

VIII. RISKS, ETHICS, AND GOVERNANCE

A. Privacy and Sensitive Inference

Deep behavioural tracking risks inferring sensitive user attributes (e.g., health status, income, religion). Governance

state masks prevent sensitive features from entering agent prompts.

B. Manipulation and Dark Patterns

Adaptive UI must not exploit cognitive biases (e.g., artificial scarcity countdowns or hidden fees). Policy guardrails enforce strict dark-pattern prohibition rules.

C. Bias and Unequal Service

Recommenders can marginalize niche products or segment users unfairly. Periodic bias audits ensure equitable service quality across user demographics.

D. Hallucination and Catalogue Grounding

Generative content modules must be strictly grounded in verified database attributes. Unverified claims immediately trigger fallback to standard product descriptions.

E. Operational Reliability

Fallback mechanisms guarantee uninterrupted service during LLM timeouts, API outages, or unexpected model responses.

IX. DISCUSSION

The central design principle of this framework is the explicit boundary between proposal generation and execution authority. LLMs excel at contextual reasoning and creative adaptation proposals, but struggle with rigid constraint enforcement. By assigning policy guardrails to a deterministic orchestrator, platforms capture generative capabilities without compromising stability, safety, or legal compliance.

X. LIMITATIONS AND FUTURE RESEARCH

This paper presents a conceptual reference architecture and review framework. Limitations include the lack of live production benchmarks. Future work will focus on deploying the open-source reference prototype, conducting user studies, and benchmarking offline policy promotion algorithms.

XI. CONCLUSION

Personalization in modern e-commerce requires unifying recommendation algorithms, digital twin state, generative interfaces, and safe policy learning. The proposed reference architecture demonstrates how multi-agent collaboration and declarative UI generation can be safely deployed under deterministic policy guardrails. Offline-first learning ensures continuous improvement while mitigating operational risk, establishing a blueprint for next-generation autonomous retail platforms.

ACKNOWLEDGMENT

The authors thank the Department of Computer Engineering, Sandip Institute of Technology and Research Centre (SITRC), Nashik, for providing academic support and research facilities, and express sincere gratitude to Prof. Devendra Singh

Kushwaha for invaluable guidance and supervision throughout this research.

XII REFERENCE

- [1] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in Proc. ACM Symp. User Interface Softw. Technol. (UIST), 2023.
- [2] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in Proc. Int. Conf. Learn. Represent. (ICLR), 2023.
- [3] Q. Wu et al., "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," arXiv preprint arXiv:2308.08155, 2023.
- [4] G. Li et al., "CAMEL: Communicative agents for 'mind' exploration of large scale language model society," in Proc. NeurIPS, 2023.
- [5] S. Hong et al., "MetaGPT: Meta programming for multi-agent collaborative framework," arXiv preprint arXiv:2308.00352, 2023.
- [6] C. Qian et al., "ChatDev: Communicative agents for software development," arXiv preprint arXiv:2307.07924, 2023.
- [7] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," in Proc. NeurIPS, 2023.
- [8] G. Wang et al., "Voyager: An open-ended embodied agent with large language models," arXiv preprint arXiv:2305.16291, 2023.
- [9] T. Guo et al., "Large language model based multi-agents: A survey of progress and challenges," arXiv preprint arXiv:2402.01680, 2024.
- [10] X. Liu et al., "AgentBench: Evaluating LLMs as agents," in Proc. Int. Conf. Learn. Represent. (ICLR), 2024.
- [11] S. Zhou et al., "WebArena: A realistic web environment for building autonomous agents," in Proc. ICLR, 2024.
- [12] T. Xie et al., "OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments," arXiv preprint arXiv:2404.07972, 2024.
- [13] C. Rawles et al., "AndroidWorld: A dynamic benchmarking environment for autonomous agents," arXiv preprint arXiv:2405.14570, 2024.
- [14] T. Schick et al., "Toolformer: Language models can teach themselves to use tools," in Proc. NeurIPS, 2023.
- [15] B. Wang, G. Li, and Y. Li, "Enabling conversational interaction with mobile UI using large language models," in Proc. CHI, 2023.
- [16] S. M. Wasti, K. Q. Pu, and A. Neshati, "Large language user interfaces: Voice interactive user interface synthesis," in Proc. CUI, 2024.
- [17] OpenAI, "GPT-4 technical report," arXiv preprint arXiv:2303.08774, 2023.
- [18] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," arXiv preprint arXiv:2307.09288, 2023.
- [19] S. Geng, S. Liu, Z. Fu, Y. Ge, and Y. Zhang, "Recommendation as language processing: A unified framework for recommendation," in Proc. RecSys, 2022.
- [20] K. Bao et al., "TALLRec: An effective and efficient tuning framework to align large language models with recommendation," in Proc. RecSys, 2023.
- [21] L. Wang et al., "User behavior simulation with large language model based agents," arXiv preprint arXiv:2306.02552, 2023.
- [22] J. Lin et al., "How can recommender systems benefit from large language models: A survey," arXiv preprint arXiv:2306.05817, 2023.
- [23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.
- [24] L. Ouyang et al., "Training language models to follow instructions with human feedback," in Proc. NeurIPS, 2022.
- [25] Kushwaha, D. S., Shaikh, S. M., Patil, S. S., Patil, V. V., & Puranchand, S. P. (2025). AI-Powered Productivity Assistant for Students and Professionals. International Journal on Advanced Computer Engineering and Communication Technology, 14(1), 653–657. <https://doi.org/10.65521/ijacect.v14i1.739>
- [26] Devendra Singh Kushwaha, MalavikaSunilKumarNair, Tanishka Deepak Takate, Amruta Manish Kokate, & Yogita Gautam Wagh. (2025). Development of Trust-Weighted Reinforcement Learning SystemforPersonalized Financial Advisory Services. International Journal on Advanced Computer Engineering and Communication Technology, 14(1), 663–665. <https://doi.org/10.65521/ijacect.v14i1.744>

[27] Kushwaha, D., Nair, M., Wagh, Y., Takate, T., & Kokate, A. (2026). Survey on Intelligent Trust-Weighted Reinforcement Learning for Financial Advisory Systems. *International Journal on Advanced Computer Engineering and Communication Technology*, 15(1), 15–20. <https://doi.org/10.65521/ijacect.v15i1.1717>

[28] P. K. Mannepal, D. Singh Kushwaha, S. Kalamdhar, V. Nagrale and V. Rajpoot, "Face Recognition Based on Cascade Classifier Using Deep Learning," 2023 1st International Conference on Innovations in High Speed Communication and Signal Processing (IHCSP), BHOPAL, India, 2023, pp. 63-68, doi: 10.1109/IHCSP56702.2023.10127172.

[29] Ganesh R. Kadam, Narendra U. Jadhav, Md Abdul Wassay, Yash R. Mohabe, Kaushal B. Loharkar, Devendra Singh Kushwaha" Facial Recognition-Based Classroom Attendance System with Real-Time Group Photo Processing Using Machine Learning Approach" *Proceedings of International Conference on Computer Science and Communication Engineering (ICCSCE 2025)*, Atlantis Press, p:1166-1177, doi:10. 2991/978-94-64 63-8 58-5_97

[30] Pankaj Kawadkar, Deepak Kumar Rathore, Devendra Singh Kushwaha, B. Rebecca, G Priyanka Jeeva Karunya, P. M. Yohan; Performance analysis of image compression techniques for MRI image using machine learning techniques. *AIP Conf. Proc.* 4 November 2024; 3214 (1): 020038. <https://doi.org/10.1063/5.0 2391 27>

[31] Y. Qamsane et al., "A methodology to develop and implement digital twin solutions for manufacturing systems," *IEEE Access*, vol. 9, pp. 44247-44265, 2021.

[32] Dhanalakshmi, P., et al. "Digital Twins and In Silico Disease Models for Personalized Gene and Cell Therapy." In *AI-Driven Innovations in Genome, Epigenome, and Mitochondrial Therapies*, edited by Divya Andy, et al., 299-324. Hershey, PA: IGI Global Scientific Publishing, 2027. <https://doi.org/10.4018/979-8-2600-0161-5.ch010>