International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 9, Issue 3, May- June 2026 ||

DOI: 10.15680/IJCTECE.2026.0903001

Auto Agent: A Self-Orchestrating System for Autonomous Agent Creation and Execution

Mr.K Raghul[1], K Janakiraman[2], M Loga Ayyanar[3], J Ajay[4]

Assistant Professor, Department of Information Technology, Jaya Engineering College, Anna University, Chennai, Tamil Nadu, India[1]

UG Students, Department of Information Technology, Jaya Engineering College, Anna University, Chennai, Tamil Nadu, India[2,3,4]

Publication History: Received: 04.04.2026; Revised: 30.04.2026; Accepted: 03.05.2026; Published: 07.05.2026.

ABSTRACT: Auto Agent is an AI-driven autonomous execution framework designed to convert high-level natural language instructions into dynamically generated, task-specific autonomous agents capable of executing complex, multi-step workflows with minimal human intervention. The system addresses the rigidity and manual overhead of traditional rule-based and no-code automation platforms by introducing intelligent intent understanding, contextual reasoning, and adaptive execution powered by Large Language Models (LLMs). Auto Agent analyzes user input to extract goals, constraints, and dependencies, decomposes complex objectives into structured sub-tasks, and instantiates autonomous agents on demand that can reason, plan, and act independently. The framework incorporates a modular architecture consisting of intent analysis, planning and reasoning, tool selection, execution control, and feedback-driven memory, enabling reliable orchestration of APIs, services, and computational tools in real time. Through continuous validation, adaptive decision-making, and execution optimization, Auto Agent enhances robustness, scalability, and fault tolerance across diverse operational environments. By significantly reducing the need for manual workflow design and static configurations, the proposed system improves productivity, accelerates automation deployment, and establishes a scalable foundation for next-generation intelligent automation systems applicable to enterprise operations, developer platforms, and personal AI-driven task management

KEYWORDS : Autonomous Agents, Artificial Intelligence, Large Language Models (LLMs), Intelligent Automation, Natural Language Processing (NLP), Task Decomposition, AI-Based Execution, Tool Orchestration, Agent-Based Systems, Workflow Automation ,Agentic AI ,GEN AI

I. INTRODUCTION

Autonomous systems are rapidly reshaping the landscape of software engineering, driven by advances in artificial intelligence, large language models (LLMs), and distributed computing. Despite these advancements, most existing AI-powered tools remain assistive rather than autonomous, requiring continuous human input, supervision, and orchestration. Users must manually define workflows, integrate tools, write scripts, and monitor execution, resulting in significant cognitive overhead and inefficiencies. This limitation becomes more evident in complex, multi-step tasks that involve reasoning, decision-making, tool usage, and adaptive execution across dynamic environments.

The emergence of agentic AI introduces a paradigm shift from passive assistance to proactive autonomy. Intelligent agents are capable of perceiving user intent, decomposing high-level goals into executable subtasks, selecting appropriate tools, and iteratively refining their actions based on feedback and environmental context. However, current implementations of agent-based systems are often fragmented, domain-specific, and lack a unified framework for dynamic agent creation, orchestration, and lifecycle management. This fragmentation restricts scalability, interoperability, and real-world applicability.

To address these challenges, this project proposes Auto Agent: An Intelligent System for On-Demand Creation of Task-Specific Autonomous Agents, a novel framework that transforms natural language instructions into fully functional, goal-oriented autonomous agents. Unlike traditional AI systems, Auto Agent dynamically generates agents tailored to specific user requests, enabling end-to-end task execution without continuous human intervention. The system leverages advanced LLMs for intent understanding, modular toolchains for execution, and adaptive planning mechanisms for real-time decision-making.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 9, Issue 3, May- June 2026 ||

DOI: 10.15680/IJCTECE.2026.0903001

II. LITREATURE SURVEY

Zozan Keskin et al.[1] proposed an LLM-enhanced human–machine interaction framework for manufacturing environments, where users can interact with complex industrial data through natural language. The system integrates a modular multi-agent architecture with components such as anomaly detection, SQL agents, and visualization agents, coordinated by a central orchestrator to deliver context-aware and user-friendly insights.

Lei Wang et al. [2] presented a comprehensive survey on LLM-based autonomous agents, highlighting how they overcome limitations of traditional rule-based systems. The study introduces a unified architecture consisting of profiling, memory, planning, and action modules, enabling agents to perform reasoning, task decomposition, and adaptive decision-making in dynamic environments.

Carlo Adornetto et al. [3]introduced generative agent-based models (GABMs), which enhance traditional simulations by incorporating LLM-powered agents capable of memory, reflection, and natural language interaction. These agents exhibit more realistic, human-like behavior, though the authors recommend hybrid approaches to balance flexibility with interpretability.

Ruofan Lu et al. [4]analyzed why autonomous agents fail in real-world task execution, identifying three major failure categories: planning, execution, and response generation. Their evaluation shows that current systems achieve only about a 50% success rate, emphasizing the need for improvements such as feedback-driven planning and self-correction mechanisms.

Qian Wang et al. [5]introduced MegaAgent, a large-scale LLM-based multi-agent system that eliminates reliance on predefined workflows. It uses a hierarchical architecture with a central “Boss Agent” that dynamically decomposes tasks and coordinates multiple agents for parallel execution, achieving higher scalability, flexibility, and performance compared to existing frameworks.

III. PROBLEM STATEMENT

Despite significant progress in artificial intelligence and automation, existing systems largely function as assistive tools rather than autonomous executors. Current AI solutions require users to explicitly define workflows, select tools, provide step-by-step instructions, and continuously monitor execution. This results in substantial manual effort, cognitive load, and inefficiency, particularly when dealing with complex, multi-stage tasks. Modern applications such as software development, data processing, web automation, and system operations often involve heterogeneous tools, dynamic environments, and iterative decision-making. However, there is no unified system capable of understanding high-level user intent and autonomously translating it into structured, executable workflows. Users are forced to bridge this gap manually, leading to increased errors, time consumption, and dependency on technical expertise.

Additionally, existing agent-based systems suffer from critical limitations:

- Lack of dynamic agent generation tailored to unique user requests
- Poor interoperability across tools, APIs, and platforms
- Limited context awareness and adaptive reasoning
- Absence of collaborative multi-agent coordination for complex tasks
- Inadequate memory and learning mechanisms for continuous improvement
- Concerns around security, reliability, and execution control

These shortcomings prevent current systems from achieving true autonomy and scalability. As a result, users cannot fully leverage AI to automate end-to-end processes, and the vision of intelligent digital agents acting independently on behalf of users remains largely unrealized.

Therefore, there is a critical need for a system that can automatically create, orchestrate, and manage task-specific autonomous agents from natural language inputs, enabling seamless execution of complex tasks with minimal human intervention while ensuring adaptability, scalability, and security.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 9, Issue 3, May- June 2026 ||

DOI: 10.15680/IJCTECE.2026.0903001

IV. RESEARCH METHODOLOGY

4.1 System Architecture

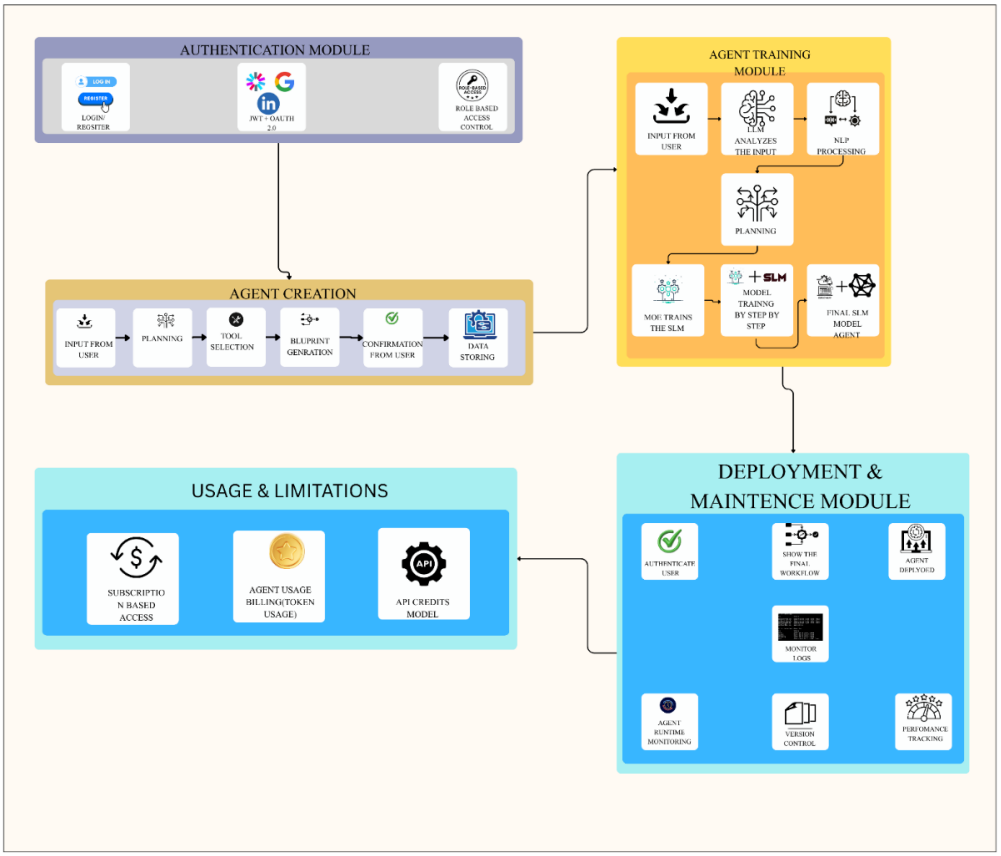

Fig 1 System Architecture

Description

The Auto Agent architecture is designed as a modular and scalable system that enables users to automatically create, train, and deploy task-specific autonomous agents using natural language instructions. The system begins with the Authentication Module, which ensures secure access through login, registration, and third-party authentication methods such as Google or LinkedIn. Role-based access control mechanisms regulate user permissions and system interactions. Once authenticated, users interact with the Agent Creation Module, where they provide task descriptions or instructions. The system analyzes the input, generates a planning structure, selects the appropriate tools or APIs, and automatically constructs a workflow for the agent. Users can review and confirm the configuration before the system finalizes memory settings and prepares the agent for training.

The Agent Training Module transforms the planned workflow into a functional AI agent. User input is processed through a Large Language Model (LLM), which performs intent analysis and natural language processing to extract relevant tasks and objectives. Based on this analysis, a planning engine decomposes the task into smaller operational steps. The system then trains or configures a Small Language Model (SLM) that specializes in executing the defined workflow efficiently. Training occurs in iterative steps, allowing the model to refine its responses and behavior according to the agent's purpose. After successful training and validation, the final SLM-based agent model is produced, optimized for performance, accuracy, and lightweight deployment. Once the agent is trained, it moves into the Deployment and Maintenance Module, where it is activated for real-world usage. This module manages agent configuration, workflow execution, monitoring, and version control to ensure reliability and maintainability over time.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 9, Issue 3, May- June 2026 ||

DOI: 10.15680/IJCTECE.2026.0903001

The **Auto Agent system** is designed using a modular architecture where each module is responsible for a specific functionality in the agent lifecycle. Based on the system architecture, the platform is divided into the following major modules:

- Authentication Module
- Agent Creation Module
- Agent Training Module
- Deployment and Maintenance Module
- Usage and Limitations Module

Each module performs a dedicated role within the system, starting from user authentication and agent creation to training, deployment, monitoring, and usage management. These modules work together to provide a complete environment for automatically creating and operating task-specific autonomous agents.

Authentication Module

The **Authentication Module** serves as the security gateway of the Auto Agent platform and is responsible for verifying user identity, managing user accounts, and enforcing controlled access to system resources. Since the Auto Agent system allows users to create, deploy, and manage autonomous AI agents, protecting access to these capabilities is essential. This module ensures that only verified and authorized users are able to interact with the system. It manages the complete lifecycle of user authentication, including account registration, login, session management, role assignment, and secure logout. By establishing a secure authentication framework, the module prevents unauthorized access and protects sensitive operations performed within the system.

Agent Creation Module

The Agent Creation Module is responsible for transforming user instructions into a functional autonomous agent within the Auto Agent platform. When a user submits a task description, the system processes the instruction and analyzes it to understand the user's objective. This module uses natural language processing techniques to interpret the instruction, identify the task goals, and determine the required actions that the agent must perform. By analyzing the user input, the system converts human-readable instructions into structured machine-understandable information.

Agent Training Module

The **Agent Training Module** is responsible for converting the generated agent plan into a trained and optimized AI model capable of performing the assigned tasks. After an agent is created, the system processes the user instruction and workflow to train the agent so that it can execute tasks efficiently. This module analyzes the structured workflow produced during the agent creation stage and prepares the training pipeline required for the agent to learn how to perform the required operations.

During this process, the system uses **Natural Language Processing (NLP)** and **Large Language Model (LLM) analysis** to understand the task objectives and convert them into structured machine-learning operations. The training process may involve refining task execution logic, adjusting model parameters, and validating the performance of the agent using test cases. Once the training process is completed successfully, the system produces a **final Small Language Model (SLM) agent**, which is optimized for executing the specific task defined by the user.

Deployment and Maintenance Module

The Deployment and Maintenance Module is responsible for activating the trained agent and managing its lifecycle within the Auto Agent system. After the agent is created and trained, it must be deployed so that it can perform tasks in a real execution environment. This module ensures that the agent is properly configured, initialized, and integrated with the required system services before execution begins. It also manages the operational workflow of the agent and ensures that the agent functions correctly according to the defined task plan.

In addition to deployment, this module also handles continuous monitoring, maintenance, and performance management of the deployed agents. The system tracks agent activity, execution logs, and performance metrics to ensure that the agent operates efficiently. Features such as agent monitoring, version control, and performance tracking allow administrators and users to manage updates and improvements without disrupting the system. This module ensures that deployed agents remain reliable, maintainable, and scalable within the Auto Agent platform.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 9, Issue 3, May- June 2026 ||

DOI: 10.15680/IJCTECE.2026.0903001

Usage and Limitations Module

The Usage and Limitations Module is responsible for controlling system resource usage and managing operational limits within the Auto Agent platform. Since autonomous agents rely on computational resources such as API calls, processing power, and token consumption, it is important to regulate how these resources are used. This module ensures that users operate within predefined limits based on their subscription plans or usage policies. It helps maintain system stability and prevents excessive consumption of resources by any single user. The core idea behind MegaAgent is a hierarchical, operating system–inspired architecture. A central Boss Agent breaks down high-level tasks into subtasks and delegates them to Admin Agents, which can further spawn additional agents as needed. This creates dynamic, multi-level agent groups capable of parallel execution, significantly improving efficiency and scalability. Unlike static systems, this structure allows the system to adapt its organization based on task complexity.

A major innovation is its multi-level monitoring mechanism. Individual agents track their progress using internal checklists, Admin Agents supervise sub-agents at the group level, and the Boss Agent performs system-level validation. This layered oversight reduces hallucinations and prevents error propagation across the workflow. The system also includes advanced communication and memory mechanisms, such as asynchronous message queues and vector-based memory storage, enabling both short-term coordination and long-term contextual recall.

V. RESULT

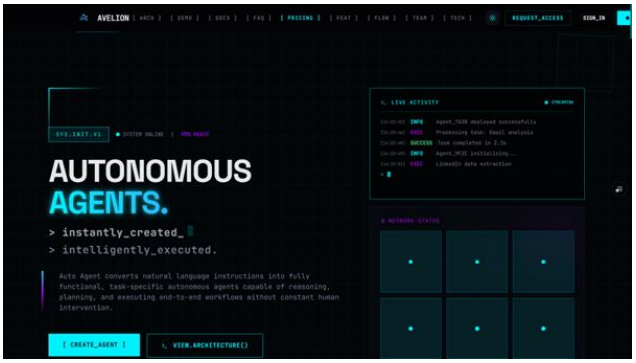

Fig 2 Home Page

The home page serves as the entry point of the system, providing an overview of autonomous agents and key functionalities. It highlights navigation options and introduces users to the platform's purpose.

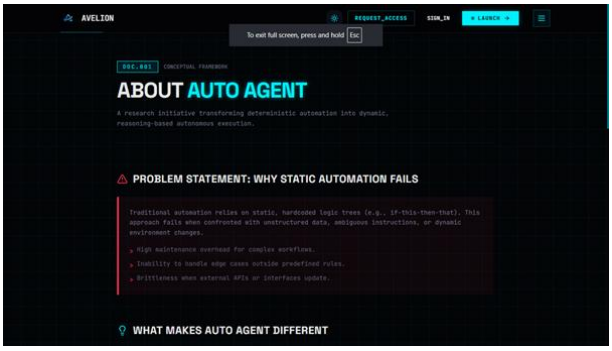

Fig 3 About Page

This page explains the concept of the Auto Agent system, including its objectives, working principles, and how it leverages AI to automate task execution.

Fig 4 Features Page

The features page showcases the core capabilities of the system, such as agent creation, task automation, and intelligent decision-making, giving users a clear understanding of what the platform offers.

Fig 5 Features Page

This page allows users to securely log into the system. It ensures authentication before accessing personalized features and agent functionalities.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 9, Issue 3, May- June 2026 ||

DOI: 10.15680/IJCTECE.2026.0903001

Fig 6 User DashBoard

The dashboard provides a centralized interface where users can view their activities, manage agents, and monitor ongoing tasks in a structured manner.

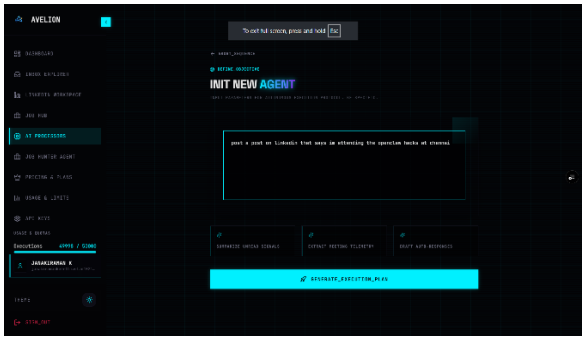

Fig 7 Prompt Input Page

This page enables users to input natural language prompts. The system interprets these prompts and converts them into task-specific autonomous agents.

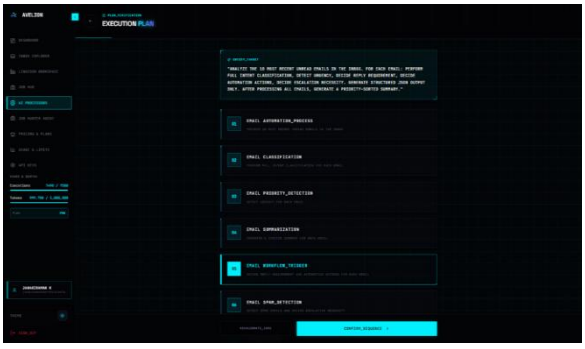

Fig 8 Blue Print Generation Page

Here, the system generates a structured blueprint of the task based on the user's input. It outlines the steps and logic required for execution.

Fig 9 Agent WorkFlow Page

This page visually represents how different agents interact and execute tasks. It shows the workflow, dependencies, and coordination between agents.

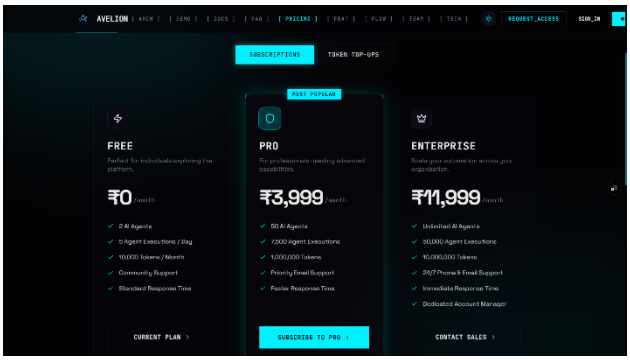

Fig 10 Price Plan Page

The pricing page displays available subscription plans, helping users choose a plan based on their usage needs and features required.

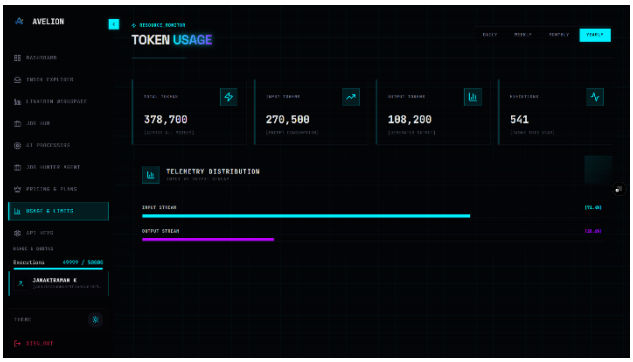

Fig 11 Token Usage Tracking Page

This page tracks system usage in terms of tokens or computational resources, allowing users to monitor consumption and optimize usage.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 9, Issue 3, May- June 2026 ||

DOI: 10.15680/IJCTECE.2026.0903001

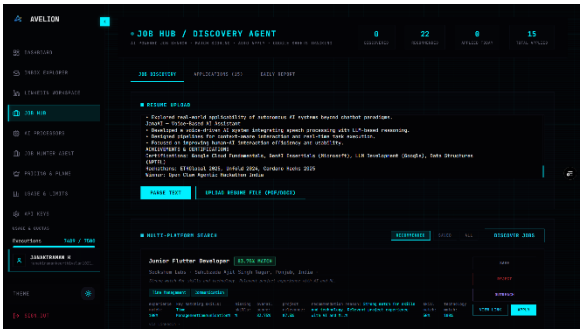

Fig 12 Job Application Page

This page allows users to submit applications or tasks to the system, initiating automated processing by agents.

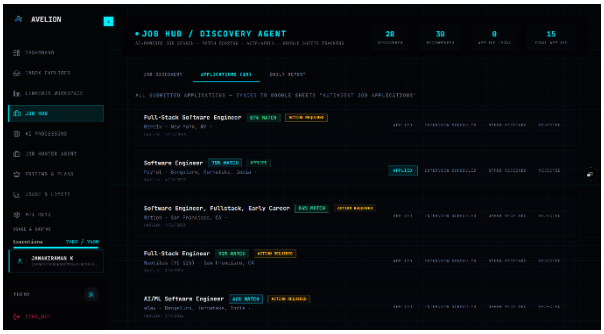

Fig 13 List Of Applications Of Job Page

This page displays all submitted jobs or applications, along with their status, progress, and results for easy tracking and management.

VI. CONCLUSION AND FUTURE ENHANCEMENT

The Auto Agent system presents an intelligent and scalable approach to automating task execution through the dynamic creation of task-specific autonomous agents. By leveraging advancements in artificial intelligence, natural language processing, and modular architecture, the system effectively converts user-provided natural language input into executable workflows. Core modules such as authentication, agent creation, training, and deployment are designed in a structured manner, ensuring seamless interaction and efficient system performance. A key strength of the system lies in its ability to interpret user intent and generate optimized execution plans, reducing the need for technical expertise and making it accessible to a broader audience. Extensive testing across unit, integration, system, and performance levels confirms the system's reliability, robustness, and scalability. The modular design further enables flexibility, allowing the platform to adapt to various domains such as workflow automation and intelligent task management, thereby establishing a strong foundation for AI-driven autonomous systems.

Despite achieving its objectives, the Auto Agent system offers significant scope for future enhancements to improve its capabilities and real-world applicability. The integration of more advanced and fine-tuned large language models can enhance accuracy in understanding, planning, and agent generation, while reinforcement learning can enable agents to learn and improve over time. Expanding the system to support multi-modal inputs such as voice, images, and real-time data can increase interactivity and usability. Incorporating Web3 technologies, including blockchain-based identity and decentralized storage like IPFS, can enhance security, transparency, and ownership. Scalability can be further strengthened through cloud deployment, containerization, and microservices architecture, enabling efficient handling of large-scale operations. Additional improvements such as visual agent builder interfaces, advanced security mechanisms like multi-factor authentication and anomaly detection, real-time analytics dashboards, and integration with external APIs and enterprise tools can significantly expand the system's functionality and adoption across diverse industries.

REFERENCES

1. Mojtaba A. Farahani, Md Irfan Khan, Thorsten Wuest, "Hybrid Agentic AI and Multi-Agent Systems in Smart Manufacturing", Journal of Manufacturing Systems (Elsevier), ISSN: 0278-6125, Volume: 72, Page: 1–15, Year: 2026
2. Leon Staufer, Kevin Feng, Kevin Wei et al., "The 2025 AI Agent Index: Documenting Technical and Safety Features of Deployed Agentic AI Systems", arXiv / MIT, Page: 1–40, Year: 2026
3. Bin Xu, "AI Agent Systems: Architectures, Applications, and Evaluation", ACM Computing Surveys, ISSN: 0360-0300, Volume: 58, Page: 1–35, Year: 2026
4. Jiacheng Miao, Joe R. Davis, James Zou, "Paper2Agent: Reimagining Research Papers As Interactive and Reliable AI Agents", arXiv / Stanford, Page: 1–28, Year: 2025
5. Naveen Krishnan, "AI Agents: Evolution, Architecture, and Real-World Applications", Springer AI Review, ISSN: 2524-7562, Volume: 9, Page: 1–25, Year: 2025

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 9, Issue 3, May- June 2026 ||

DOI: 10.15680/IJCTECE.2026.0903001

6. Yue Liu, Sin Kit Lo, Qinghua Lu, Liming Zhu, "Agent Design Pattern Catalogue for Foundation Model Based Agents", Journal of Systems and Software (Elsevier), ISSN: 0164-1212, Issue: 2, Volume: 210, Page: 111–130, Year: 2025
7. Ajay Bandi, Bhavani Kongari et al., "The Rise of Agentic AI: Definitions, Architectures and Challenges", Future Internet (MDPI), ISSN: 1999-5903, Issue: 3, Volume: 17, Page: 1–30, Year: 2025
8. Shuai Yao, Jie Zhao, Dong Yu et al., "ReAct: Synergizing Reasoning and Acting in Language Models", ICLR (IEEE indexed), Page: 1–20, Year: 2024
9. Ken Huang, "Agentic AI: Theories and Practices", Springer Nature, Page: 1–300, Year: 2025
10. Thorsten Wuest et al., "Agentic AI for Prescriptive Maintenance Systems", IEEE Transactions on Industrial Informatics, ISSN: 1551-3203, Issue: 4, Volume: 21, Page: 2200–2210, Year: 2026
11. Michael Luck, Kate Larson, "Autonomous Agents and Multi-Agent Systems", Springer, ISSN: 1387-2532, Issue: 2, Volume: 38, Page: 1–20, Year: 2025
12. Stephen Casper et al., "Evaluating Autonomous AI Agents: Benchmarks and Safety", IEEE AI Magazine, ISSN: 0738-4602, Issue: 1, Volume: 46, Page: 45–60, Year: 2026
13. OpenAI Research Team, "GPT-based Autonomous Agents and Tool Use", ACM Digital Library, ISSN: 0001-0782, Volume: 68, Page: 1–18, Year: 2025
14. Seedha Devi, V., Nivedha, S., Harisha, V., Mol, D. R., & Janaranjini, J. R. (2026). Enhanced prediction of PCOS and PCOD using deep learning for early diagnosis and clinical risk stratification. International Journal of Advanced Research in Computer Science & Technology (IJARCST), 9(3), 783–793.
15. Seedha Devi, V., Kumar, M. D., & Kumar, C. A. (2026). Flutter-based SOS alert and location tracking application with volunteer assist and rescue. International Journal of Research and Applied Innovations (IJRAI), 9(3), 521–530. https://doi.org/10.15662/IJRAI.2026.0903003
16. Seedha Devi, V., Selvi, D., Uma Maheshwari, K., & Yuvashree, G. (2026). Food linker: A smart system for global waste reduction. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(3), 5012–5021. https://doi.org/10.15662/IJEETR.2026.0803002
17. Seedha Devi, V., Namitha, B., Divya Dharshini, J., & Livetha, K. (2026). A hybrid biometric and geo-fencing based smart attendance system. International Journal of Advanced Research in Computer Science and Technology (IJARCST), 9(3), 794–802. https://doi.org/10.15662/IJARCST.2026.0903002
18. Mathew, A. Cybersecurity 5.0: From Firewalls to Fully Autonomous Digital Protection.
19. Mathew, A. (2024). From Conversation to Command Execution: A Comparative Threat Modeling and Risk Analysis of OpenClaw and ChatGPT. Risk, 100(1).
20. Anujaa, T., Thajudeen Ali Ahamed, A. F., Baranwal, V., Thanikaiselvan, V., Subashanthini, S., Sivaranjani Devi, C., & Rengarajan, A. (2025). A lightweight multi round confusion-diffusion cryptosystem for securing images using a modified 5D chaotic system. Scientific Reports, 15(1), 31986.
21. Aashiq Banu, S., Rao, L. K., Priya, P. S., Thanikaiselvan, Hemalatha, M., Dhivya, R., & Rengarajan, A. (2025). A review of genome to chaos: exploring DNA dynamics in security. Multimedia Tools and Applications, 84(22), 24859-24886.
22. Rajasekar, M. (2024). AI-Powered Cyber-Secure Federated Learning on AWS for Next-Generation Digital Banking Analytics. International Journal of Advanced Research in Computer Science & Technology (IJARCST), 7(3).
23. Rajasekar, M., Nahar, G., Jagatheeswaran, S., Chinthamani, S. A. M., Mohammed, S. H., & Al-Hilali, A. (2024, May). Retraction Notice: The Roadmap to Classify Malware Using ML Algo Through IOT Based SN. In 2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) (pp. 1-1). IEEE.
24. Vimal, V. R., John Justin Thangaraj, S., Narayanan, L. K., Alagu Thangam, S., Loganayagi, S., & Balakrishnan, S. (2025, April). Enhanced Phishing Detection and Classification Using an Ensemble Machine Learning Approach for URL Analysis. In International Conference on Information and Communication Technology for Intelligent Systems (pp. 229-239). Singapore: Springer Nature Singapore.
25. Vimal, V. R., Jayalakshmi, D., Narayanan, L. K., Hemavathi, R., & Loganayagi, S. (2024, November). 5G-Enabled Remote Healthcare Monitoring for Improved Patient Care. In 2024 International Conference on Recent Advances in Science and Engineering Technology (ICRASET) (pp. 1-5). IEEE.
26. Anbazhagan, K. (2024). Trustworthy and Adaptive AI Systems for Enterprise Analytics Cybersecurity and Decision Optimization Using API-First and Cloud-Native Architectures. International Journal of Technology, Management and Humanities, 10(03), 65-74.
27. Anbazhagan, K., Kumar, R., Thilagavathy, R., & Anuradha, D. (2024, March). Shortest Job First with Gateway-based Resource Management Strategy for Fog Enabled Cloud Computing. In 2024 4th International Conference on Data Engineering and Communication Systems (ICDECS) (pp. 1-6). IEEE.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 9, Issue 3, May- June 2026 ||

DOI: 10.15680/IJCTECE.2026.0903001

28. Murugeshwari, B., Sabatini, S. A., Jose, L., & Padmapriya, S. (2023). Effective data aggregation in WSN for enhanced security and data privacy. arXiv preprint arXiv:2304.14654.
29. Murugeshwari, B., Jothi, D., Hemalatha, B., & Pari, S. N. (2023). Trust Aware Privacy Preserving Routing Protocol for Wireless Adhoc Network. arXiv preprint arXiv:2304.14653.
30. Praveena, M., Saravanan, M., & Yerra, R. (2025, June). PSO MPPT based Control Framework for Photovoltaic Systems to enhance Power Quality. In 2025 5th International Conference on Intelligent Technologies (CONIT) (pp. 1-5). IEEE.
31. Vani, S., Malathi, P., Ramya, V. J., Sriman, B., Saravanan, M., & Srivel, R. (2024). An efficient black widow optimization-based faster R-CNN for classification of COVID-19 from CT images. Multimedia Systems, 30(2), 108.
32. Sugumar, R. (2025). Cyber-Secure Cloud Architecture Integrating Network and API Controls for Risk-Aware SAP Healthcare Data Platforms. International Journal of Humanities and Information Technology, 7(4), 53-60.
33. Sugumar, R. (2025). Secure and Explainable AI Systems in Cloud-Based Applications: Bridging Trust and Performance. International Journal of Engineering & Extended Technologies Research (IJEETR), 7(4), 10328-10335.
34. Anand, L., Tyagi, R., & Mehta, V. (2024, January). Food recognition using deep learning for recipe and restaurant recommendation. In Proceedings of Eighth International Conference on Information System Design and Intelligent Applications (pp. 269-279). Singapore: Springer Nature Singapore.
35. Anand, L. (2024). AI-Powered Cloud Cybersecurity Architecture for Risk Prediction and Threat Mitigation in Healthcare and Finance. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 7(Special Issue 1), 5-12.