Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

ILMIX AI: A Context-Aware Voice Assistant for Personalized Task Automation

Imran[1], Gopal K[2], Deepan S U[3], Akash M[4], Kabirdosar S[5]

[1,2,3,4,5] Department of Computer Science and Engineering, The Kavery Engineering College, Salem, Tamil Nadu, India

Abstract
Modern voice assistants are often command driven, cloud dependent, and weakly personalized, which limits privacy and offline task automation. This paper presents ILMIX AI, a context-aware voice assistant for personalized desktop automation. A modular Python prototype was developed with dynamic context weighting, confidence-gated execution, behavioral anomaly monitoring, multi-user profile separation, and local-first storage. Evaluation on desktop tasks showed 0.93 intent accuracy, 0.91 command success, 1.6 s offline response time, and fewer execution errors through clarification.

Keywords: Context-Aware Voice Assistant, Personalized Automation, Offline AI, Continual Learning, Behavioral Security

1. Introduction

Voice-based interaction has become an important modality in human-computer interaction. It supports application launch, reminder creation, file access, and routine system control while reducing dependence on keyboard and mouse input. However, many deployed assistants remain reactive and command centric. They often rely on cloud services for speech processing and reasoning. Consequently, offline reliability, privacy, and personalization remain limited in practical desktop environments.

The primary problem addressed in this work is the limited integration of context reasoning, adaptive personalization, secure multi-user handling, and safe offline automation in current voice assistants. Long dialogues may suffer from context drift, while ambiguous requests can lead to unsafe actions. In shared-device settings, preferences and recommendations may also leak across users. These issues reduce trust and weaken productivity in everyday use.

ILMIX AI is proposed to address these gaps through context-aware task automation. The objective is to interpret user intent using time, system state, and activity signals, and then adapt execution to the current situation. The system also learns user behavior from history, predicts likely next tasks, and improves through feedback over time. Local-first storage and offline-capable processing were adopted to strengthen privacy and reliability.

2. Literature Review

Desktop-oriented voice assistants have been implemented using speech recognition, text-to-speech (TTS), automation libraries, and conversational models. Supe et al. [1] and Kumari et al. [4] demonstrated practical productivity support through modular desktop automation. Anoliefo et al. [2] and Mishra and

Upadhyay [3] improved personalization and contextual understanding using machine learning methods. Even so, offline robustness and computational efficiency remained recurring limitations.

Privacy-preserving and scalable assistants have also been reported. Local-first processing reduced data exposure in Mishra et al. [5], although access to broad external knowledge was constrained. Cloud-native orchestration and large-model reasoning improved flexibility, but latency, setup complexity, and service dependence remained concerns [6], [7]. The present work addresses these limitations through local-first context reasoning, multi-user separation, anomaly-aware execution, and confidence-gated automation.

Study	Method	Advantage	Limitation
Supe et al. [1]	Desktop automation with conversational model	Strong productivity support	Internet dependency for speech services
Anoliefo et al. [2]	Machine learning with recommendation methods	Personalized task management	Slow cold-start adaptation
Mishra and Upadhyay [3]	LSTM/CNN-based natural language understanding	Improved contextual accuracy	High computational cost
Kumari et al. [4]	Modular Python automation assistant	Practical task automation	Noise sensitivity in recognition

Table 1: Literature Comparison

3. Methodology

ILMIX AI was designed as a modular desktop assistant for personalized task automation. The system workflow included speech capture, speech-to-text conversion, intent and entity extraction, context expansion, plugin-based action selection, execution, and feedback-based learning. Speech-to-text (STT) conversion was supported through a hybrid pipeline, while responses were generated using a text-to-speech engine. Natural language processing (NLP) was used to derive intent labels and required entities. A local SQLite database stored user profiles, command history, and feedback.

A Dynamic Context Weighting Engine (DCWE) was incorporated to reduce context drift during multi-turn interaction. Context signals included time of day, active application, recent command history, user identity, file associations, and network availability. Importance scores were computed in real time, and the strongest weighted signals were used for intent resolution. This approach allowed the assistant to prioritize current relevance over stale dialogue history.

Safety and personalization were handled through adaptive confidence thresholding, behavioral anomaly detection, and multi-user separation. A confidence score was computed from intent certainty, entity completeness, and historical success rates. When confidence was low, clarification was requested before execution. Behavioral anomaly monitoring constrained sensitive actions under unusual command patterns. Multi-user behavioral separation stored profiles locally and prevented cross-user leakage. Continual Learning Without Forgetting (CLWF) preserved stable behavior while allowing incremental personalization updates. A 5W1H context expansion step was used to infer who, what, where, when, why, and how for ambiguous requests. A zero internet mode ensured core local automation without external calls. Smart file context linking and a plugin-based modular architecture supported extensibility.

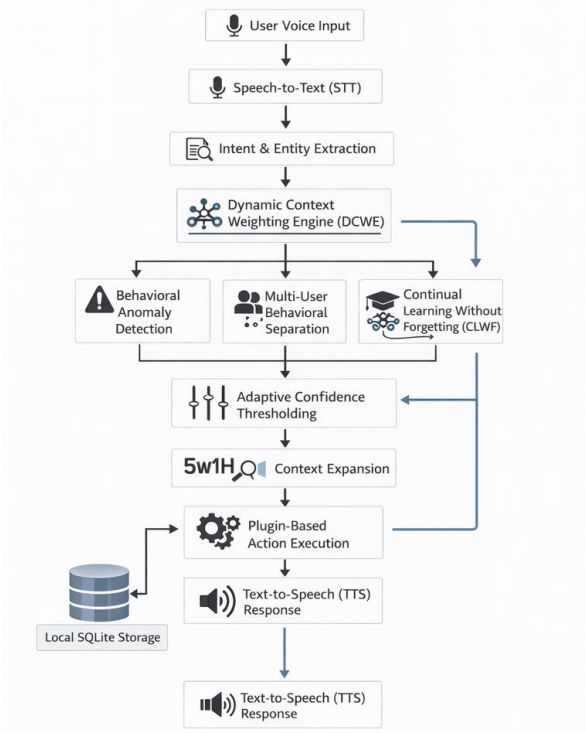

Fig 1: System Architecture Diagram

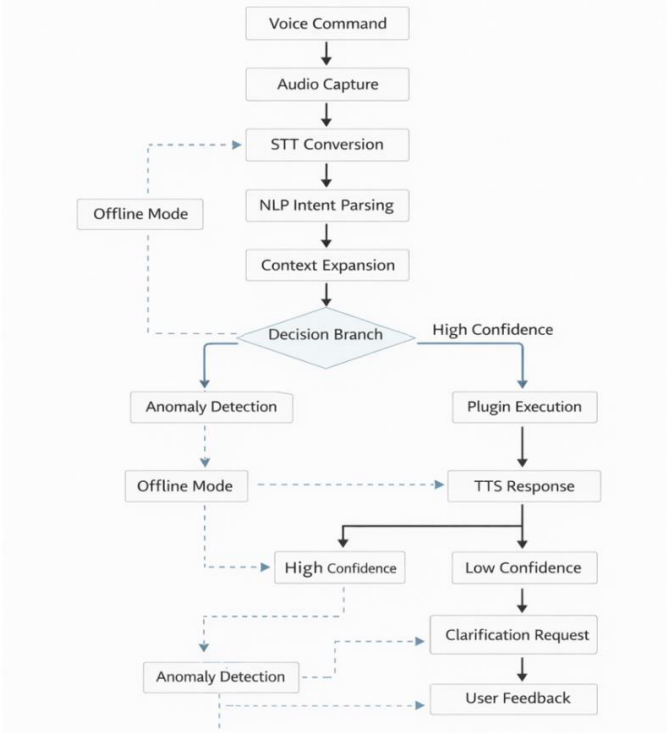

Fig 2: Workflow Diagram

4. Results and Discussion

A Python 3.x prototype of ILMIX AI was implemented using speech recognition libraries, a local TTS engine, NLP-based parsing, operating system automation, and SQLite-backed storage. The prototype was tested on representative desktop tasks such as application launching, local file search, reminder creation,

and multi-step workflows requiring contextual disambiguation. Evaluation focused on intent recognition, command success, response latency, clarification behavior, and offline reliability.

The system showed reliable execution for common desktop operations. Context weighting improved interpretation when time and system state were relevant, especially during repeated routines. Personalization reduced repeated parameter specification after sufficient interaction history had been collected. Clarification was triggered when confidence fell below the threshold, which reduced incorrect execution under ambiguous input. In such cases, post-clarification completion was consistently higher than direct low-confidence execution.

Offline behavior remained stable for local tasks because parsing, local retrieval, file operations, and system automation were executed without remote calls. As expected, open-domain knowledge retrieval was limited in offline mode to preserve privacy and reliability. Behavioral anomaly detection added an additional safety layer by increasing confirmation requirements for atypical command sequences. Overall, the prototype demonstrated improved usability for repeated desktop workflows, with the most visible gains appearing after personalization data had accumulated.

Metric	Observed Performance	Measurement Context
Intent classification accuracy	0.93	Parsed intents against labeled test commands
Command execution success rate	0.91	Desktop automation tasks with validated outcomes
Average response time online mode	1.2 s	Including STT, context expansion, and plugin selection
Average response time offline mode	1.6 s	Local processing and operating system automation
Post-clarification success rate	0.95	Successful execution after user clarification

Table 2: System Performance

Fig 3: Application Screenshot – Home Page

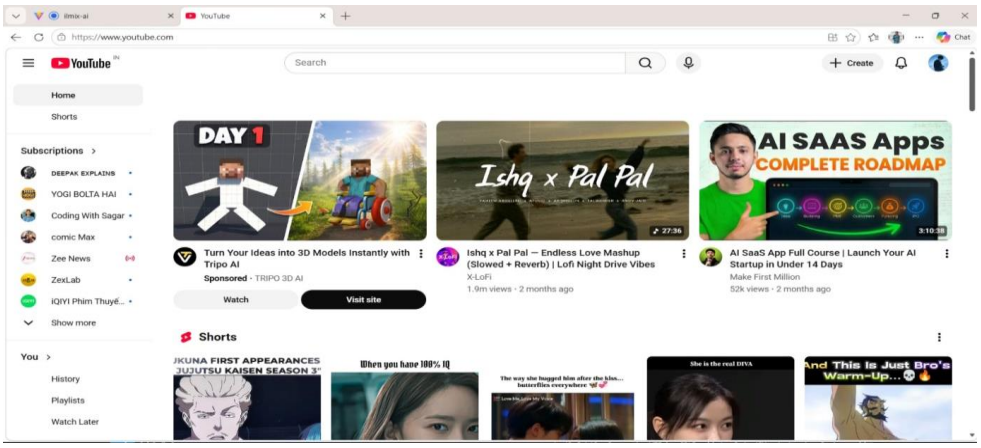

Fig 4: Application Screenshot – Results Page

5. Conclusion

ILMIX AI was presented as a context-aware voice assistant for personalized task automation. The proposed system combined dynamic context weighting, adaptive confidence gating, multi-user profile separation, anomaly-aware safeguards, continual learning, and offline-capable local processing. The prototype showed improved task success, better relevance in repeated workflows, and safer execution under uncertainty. Future work should improve noise robustness, accent generalization, and lightweight offline speech recognition while expanding plugin coverage for broader productivity use.

6. Acknowledgement

The authors acknowledge the Department of Computer Science and Engineering, ABC Institute of Technology, for providing the facilities required to develop and test the prototype. Guidance from faculty mentors and feedback from reviewers are also gratefully acknowledged.

References

1. A. Supe et al., “ARIA: An Intelligent Voice-Enabled Virtual Assistant for Desktop Automation and Conversation,” 2025.
2. C. J. Anoliefo et al., “Building an AI-Powered Virtual Assistant for Personalized Task Management,” 2025.
3. M. Mishra and S. K. Upadhyay, “AI Voice Assistant: A Scalable and Adaptable Framework for Smart Personal Assistance,” 2025.
4. R. Kumari et al., “ELLYSE: Voice Assistant with Advanced Task Automation and Personalization Using Python,” 2025.
5. M. H. Muthuraj et al., “Interactive Voice-Controlled Assistant,” 2025.
6. R. Karthick, “A Framework for Scalable AI Applications Cloud Native That Integrates Multiple Databases and Real-Time AI Orchestration,” 2025.
7. S. Chittepu et al., “Empowering Voice Assistants with Tiny Machine Learning for User-Centric Innovations and Real-World Applications,” 2025.