Asynchronous Architecture for Real-Time Interview Simulation: A Concurrent Processing Approach to AI-Driven Interviews

Parth Bhalerao[1]

[1] Duke University, Durham NC 27705, USA
*iamparthhh@gmail.com

Abstract. The traditional job interview process faces scalability, availability, and consistency challenges. As a job aspirant in the volatile job market of 2024, I identified the need to create systems that make this process more efficient and consistent for all parties involved, organizations, and candidates. With interview processes varying across companies and job seekers typically submitting hundreds of applications, preparing for each unique interview becomes daunting. Additionally, companies face scalability issues with traditional interviewing processes, where a better understanding of a larger applicant pool could significantly improve their hiring efficiency.

This paper presents an innovative interview bot that addresses the dual challenges of interview readiness and standardized evaluation. The system primarily serves as a practice platform for job aspirants seeking a more quantitative approach to their interview preparation. This platform could also help organizations standardize their interview evaluations by potentially offloading initial candidate screenings to autonomous bots capable of evaluating candidates' profiles for specific job roles.

By leveraging generative AI technologies, a comprehensive question bank, and a WebRTC framework, this platform simulates realistic interview experiences, improving interview preparedness and evaluations. The realistic simulation is achieved through dynamic question generation, detailed feedback mechanisms, AI-generated voice for natural interaction, and concurrent system design for low-latency output generation. The system incorporates industry-standard evaluation criteria to ensure optimal performance.

The body's ability to provide consistent, objective feedback while being available 24/7 addresses the limitations of traditional mock interview methods. This research contributes to the evolving landscape of AI-assisted career development tools while offering insights into scalable interview automation solutions.

Keywords: Interview Automation, Mock Interview Simulation

1 Introduction

The landscape of recruitment has fundamentally transformed in the modern digital era, with organizations conducting millions of interviews annually across global talent pools. Most of our lives have experienced technological and lifestyle advancements, including most of our work. However, the recruitment process has remained similar

S. Bhalerao et al. (eds.), *Proceedings of the International Conference on Recent Advancement and Modernization in Sustainable Intelligent Technologies & Applications (RAMSITA-2025),* Advances in Intelligent Systems Research 192,
https://doi.org/10.2991/978-94-6463-716-8_46

over decades. Similar to the earlier 80s, when a candidate shows interest in joining a company, the company employs a recruiter to evaluate the profile, then if chosen they are called in for an interview, only after which a decision is reached. This substantial commitment of human resources, coupled with the challenges of maintaining consistency across numerous interviews, presents a compelling case for automation in the initial screening phases of recruitment.

Traditional interview processes face multiple critical challenges that impact both organizations and candidates. The time-intensive nature of preliminary screenings creates bottlenecks in the hiring pipeline, often leading to delayed decisions and the potential loss of qualified candidates. Furthermore, interviewers may inadvertently introduce inconsistencies and biases into the evaluation process, affecting the fairness and reliability of candidate assessments. Organizations also struggle with standardizing evaluation metrics across different interviewers and departments, making it difficult to compare candidates objectively.

While various automated solutions have emerged in recent years, most existing systems focus primarily on scheduling automation or basic questionnaire administration. These solutions, while valuable for specific tasks, fail to address the need for dynamic, conversational interactions that can effectively assess candidate qualifications and potential. Current automated systems often lack the sophistication to adapt their questioning based on candidate responses or to evaluate subtle aspects of professional competency.

This paper presents an innovative interview bot designed to address these limitations through advanced natural language processing and adaptive questioning techniques. Our system conducts initial screening interviews while maintaining a conversational approach that closely mimics human interaction. By incorporating existing generative AI API, comprehensive evaluation frameworks, and a secure low-latency web framework, the bot provides consistent, relatively unbiased assessments while significantly reducing the time investment required from hiring managers. The primary objectives of this research include:

- Developing a scalable automated interview system that maintains consistency across candidates
- Implementing dynamic question generation that adapts to candidate responses
- Reducing the time and resources required for initial candidate screening
- Improving the candidate experience through natural, responsive interaction
- Suggesting candidate-evaluation models to assess the performance of correctly

This paper focuses mainly on implementing the project in Python, NextJs, ReactJS, and Django framework. We will also focus on implementing the system and the architecture for this platform.

2 Background

AI-enabled applications are increasingly used in numerous fields to assist the user in effectively achieving their desired outcomes. The inclusion of these AI architectures in the recruitment process is to provide optimized solutions [1] as well as beneficial for smart, intelligent digital solutions of human resource management [2,3]. AI is also used in identifying applicant perspectives in the recruitment process [4]. It is important to select appropriate candidates for a particular department and requires meticulous procedures to evaluate the capabilities of candidates. The primary step in selection involves examination and interviews [5]. Job seekers' information and their answers are correlated, and with the use of big data and predictive analytics, intentions and knowledge domains are evaluated [6]. By employing sophisticated ML algorithms coupled with the gen AI revolution, a significant improvement of 82% is obtained in the selection of the right candidate for the right position [7]. It has been noticed that the focus is on optimizing the recruitment process however, this AI agent can play an important role in predictive hiring and enhance the support to the candidates. Machine Learning applications [8] on predictive hiring highlight the use of random forest algorithms to improve performance. Software applications for interview processes based on NLP and sequence text questions are useful for identifying the right talent [10]. Large Language models are applied to generate the interview questions using the ChatGPT iterative application [11].

In short, models for optimizing the recruitment process by introducing chatbots for the interview process and shortlisting job seekers are found in the literature, especially in the last decade. However, an integrated application of voice, text, and video bots for job interviews in various domains is rarely available. Therefore, there is scope for research in aforesaid areas for the ease of job seekers as well as employers for simplifying the process.

3 System Design & Architecture

The platform is built on a microservice architecture that prioritizes scalability, real-time performance, and concurrent user support. This section details the system's core components, interactions, and the technical decisions shaping the implementation.

3.1 Architecture

The system is structured into five distinct layers, each serving specific functions while maintaining loose coupling for scalability and maintainability:

3.2 Client Layer

The client layer handles the events on the client's end such as login/logout, adding/removing interview sessions, and other pertinent actions the user intends to perform. Here are some of the key highlights for the client layer:

- The client layer's logic is handled using Next JS and React JS is used for the frontend server deployment and user interface design respectively
- The WebRTC Client helps with enabling the real-time low latency interaction for the user. This uses wss:// (Web Services Security) to transfer larger chunks of data (audio streams) efficiently to help maintain concurrency and low latency.

- Maintains the interview state management by handling and working with the cookies at the client layer.

3.3 Processing Layer

The processing of the audio stream, both transcription and the platform, is essential. This layer primarily handles communication with the other layers. The main three components of this architecture are the Django backend, Eleven Labs, and Deepgram API.
Eleven Labs API serves as another microservice in this architecture, used for creating a realistic AI voice to make the experience as true to real-world situations as possible.
Similar to Eleven Labs API, we are employing Deepgram API to create the most accurate transcriptions of the user's spoken answers. This step is critical to the functionality since transcribing the user's response is the most crucial aspect, alongside generating questions and feedback.

3.4 Application Layer

The application layer is the mission-critical layer of the system architecture (shown in Fig.1). This is the main brain of the architecture that handles the logic and the interactions with the other layers in the architecture. The main highlights of this layer are:

- This layer's main component is the Django backend which handles all the main logic of the interview, application logic (user signup, registration, other), and other web application components such as cookies management, etc.
- This implementation of the Django framework was rearchitected to be asynchronous to support the desire for low-latency
- This asynchronous implementation was possible because of the help of celery workers who keep track of the tasks at hand for the system while maintaining an efficient pipeline of tasks for a synchronous architecture like Django.

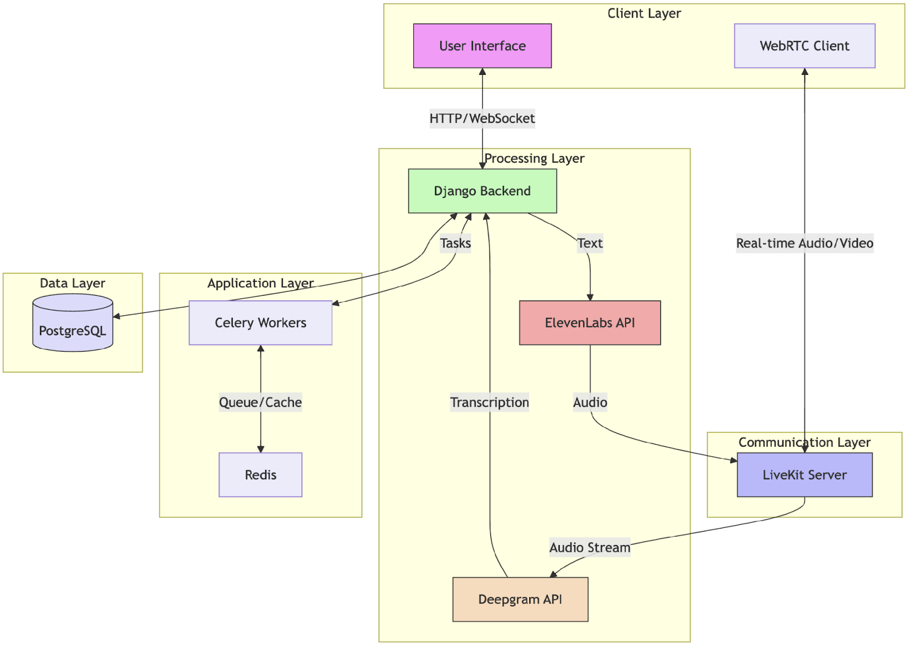

Fig. 1. System Architecture

3.5 Data Layer

This layer is where all the databases exist. We chose a serverless implementation for the Postgres server. This serverless makes the data costs relatively cheaper while ensuring that the data availability isn't compromised as much. The implementation of the databases is also branched to ensure data integrity. If ever compromised, we will have branched databases to be able to recover any data that might be critical.

3.6 System Sequence and Interaction Flow

The interview preparation platform orchestrates a complex series of interactions between multiple services to create a seamless, real-time interview experience. This section details the complete system sequence (shown in fig. 2), illustrating how different components work together to process user input, generate responses, and maintain natural conversation flow.

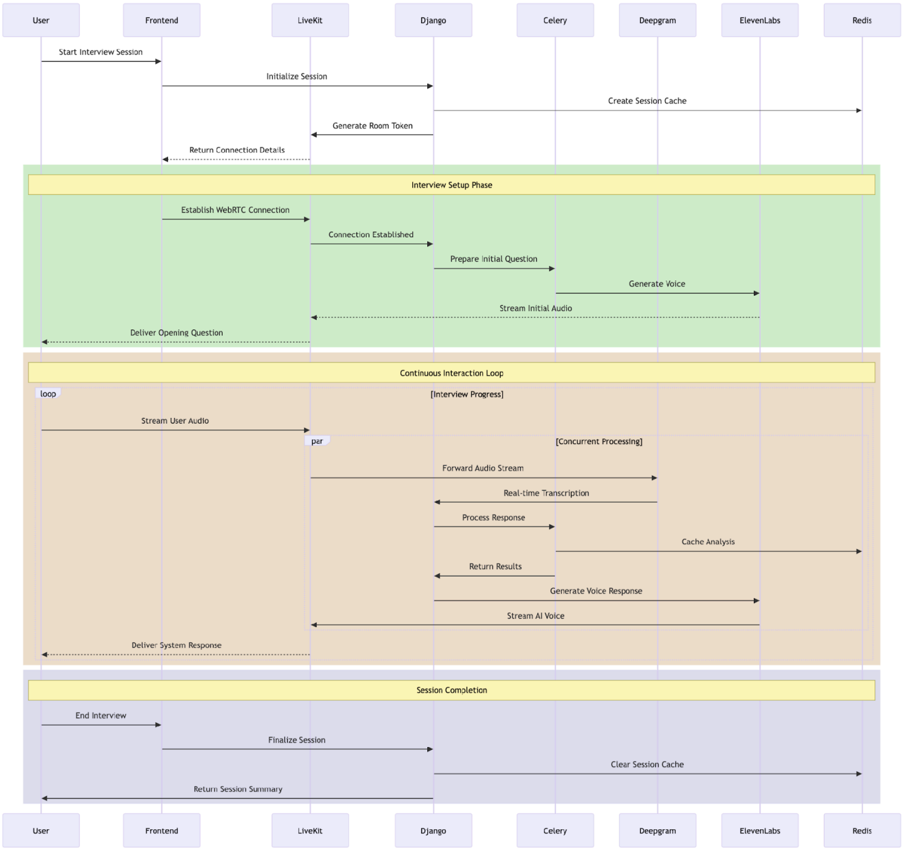

Fig. 2. Sequence Diagram for the platform

As you can see in the sequence diagram above, there are three phases to the platform sequence flow diagram.

- The interview setup phase focuses on setting up the parameters for the interview such as the role, the industry, and other parameters that control how the interview will be conducted.
- The interaction loop is continuous throughout the interview session after its initialization. This phase takes the user's audio stream file, transcribes, and shares the text spoken with the AI models to conduct the interview further and eventually create the feedback. Although the sequence in this section is relatively sequential, it's all concurrent in implementation.
- Finally, after the interview is completed, there are a lot of hygiene checks such as finishing the session, updating the status of the interview in the database, and other tasks to successfully wrap up a session are performed in this stage.

3.7 Features and Functionality

Interviews

Currently, the platform is well-equipped to handle interviews for behavioral rounds and technical rounds in data science-related fields. As the system diagrams suggest before every interview is loaded, it asks for a form to be filled that allows for the user to select the variables for the interview. Therefore, no two interviews with different parameters will have the same approach.

The platform specializes in two different types of interviews: behavioral and technical interviews.

Each one of the fields in the form above customizes the interview experience for the user (Fig. 3). Here's how the system adapts based on the variables set either by the organization or the candidate practicing for the interviews.

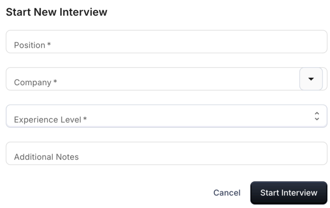

Fig. 3. Interview initiation form at client side

- Position: This field considers the position the individual is applying for. This allows the system to customize the questions for this specific role.
- Company: Some companies are part of the growing list of companies with curated evaluations based on the information we could find online on how to best prepare for these companies. For example, an interview with Google for the same position will be different from an interview for the same position at Amazon.
- Experience Level: Similarly, the experience level for the job position also defines the possibility of what kind of interview to expect.
- Additional Notes: This field is for any comments or any other information relevant to the job description that the system can use to customize the job interview experience for the candidate.

3.8 Concurrency Management & Real-time Processing

A fundamental requirement for creating an authentic interview experience was achieving real-time interaction with minimal latency. While Django serves as the core framework for our application, its synchronous nature presents challenges for handling real-time audio processing and response generation. To overcome these limitations, we implemented a sophisticated concurrency management system using Celery.

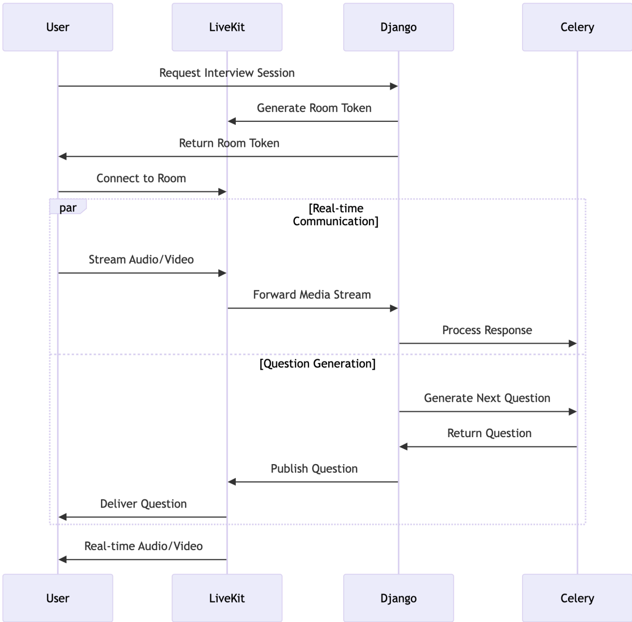

Fig. 4. Task Distribution in the Application Layer

The task distribution was a critical step for the successful implementation of the project (shown in Fig. 4). It was important to break down bigger tasks into smaller tasks that can be run asynchronously while ensuring the desired low latency. The system employs a distributed task queue architecture where:

- Django handles the core application logic and HTTP Requests
- Celery workers manage concurrent processing tasks
- Redis serves as the message broker and result backend
- Multiple tasks can be processed simultaneously without blocking the main application thread

The systems achieve this efficient queuing of the tasks to ensure concurrency by successfully breaking down systems into smaller tasks. I decided to break down the tasks into these discrete tasks:

- Audio processing and transcription
- Question generation and selection
- Response Analysis and Feedback Generation
- Performance metrics calculation

The system queues and processes each task independently, allowing for parallel execution of multiple interview sessions. This ensures the scalability that the platform demands and aims to resolve. Furthermore, what would otherwise be a highly latent process of processing audio and then generating the transcription would all be happening simultaneously. This means that the output for what is being said is generated at the same time as when the user is speaking.

3.9 Accurate Transcription

This platform is expected to be used in the wild. Therefore, it must be performed equally in a lot of varying conditions (shown in Fig. 5). The biggest challenge for this was to have a successful audio transcription for the model to work as desired.

Therefore, for this platform, I ended up choosing Deepgram's advanced transcription API which performed best for me for all the 'in the wild' tests that I performed for this platform.

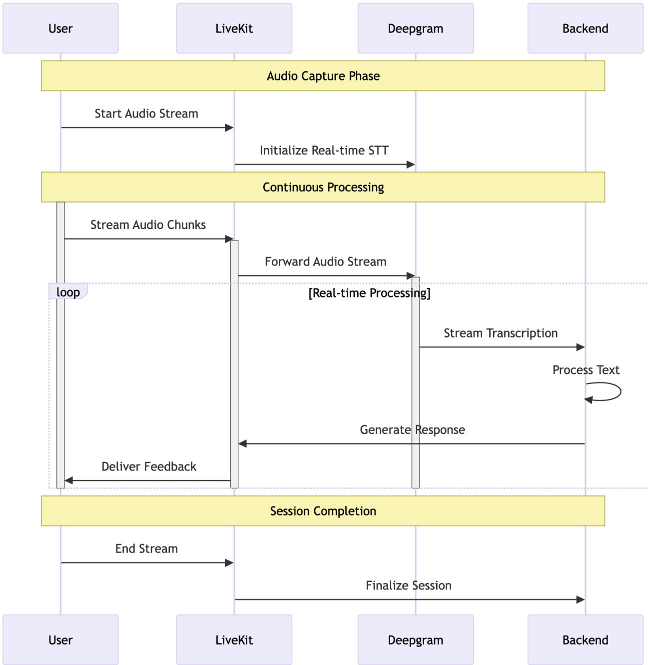

Fig. 5. Sequence diagram for accurate transcription

3.10 AI-Generated Voice

The platform implements a sophisticated voice processing pipeline that combines real-time speech recognition coupled with real-time speech generation (shown in Fig. 6). This dual-stream approach allows the interviews to feel more natural. As the inspiration for this project has been consistently to emulate the real-life interview experience, generating a realistic voice was also as critical to the whole user experience.

ADD ImAGE HERE

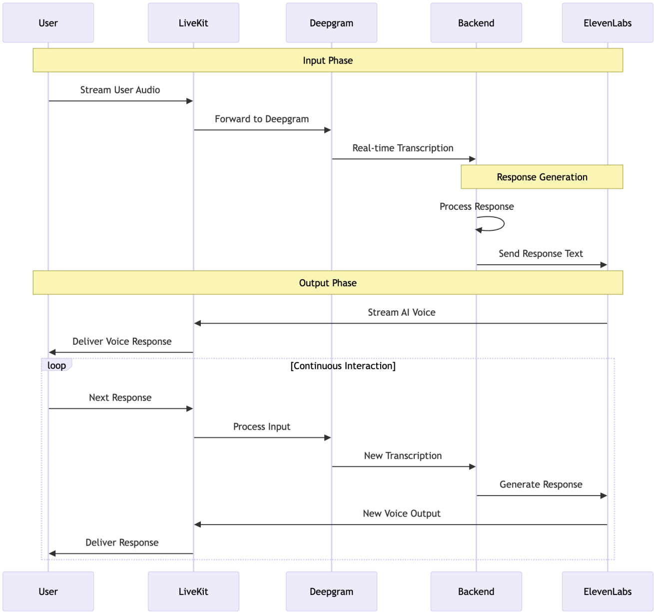

Fig. 6 Sequence Diagram for AI Voice Generation

4 Conclusion

The interview preparation platform demonstrates several significant strengths while facing certain limitations that guide our future development roadmap. This section examines these aspects comprehensively to provide a clear understanding of the system's current capabilities and areas for growth.

4.1 System Strengths

The platform excels in its technical implementation, particularly in handling real-time interactions. The integration of LiveKit for WebRTC communication ensures stable, low-latency audio streaming that the system generates a response almost instantaneously. The audio transcription coupled with the industry standard voice generation by Eleven Labs ensures that a seamless interview experience is emulated for the user.

Furthermore, how the information for the model can be expanded by allowing the organizations to further improve the capabilities by adding more questions in the question data bank, giving custom evaluation instructions, and adding more customizations to any interview experience enables this to achieve the scalability, availability, and consistency of the traditional job interview process that we aim to fix.

The system's concurrent processing architecture, built on Django and Celery, enables efficient handling of multiple simultaneous sessions without any performance degradation. This architecture proves particularly valuable for scaling the platform to accommodate increasing user demand while maintaining consistent performance.

Additionally, the experience has been designed with the user being the central focus. The platform offers accessibility through its round-the-clock availability for any candidate to practice their interviewing skills while having a quantitative approach to their interview practice.

4.2 Limitations

Despite its robust implementation, the platform faces several limitations that warrant acknowledgment. The current scope primarily covers behavioral interviews and technical interviews in data science-related fields, which, while thorough in these areas, leaves room for expansion into other domains.

The system's heavy dependence on external APIs for mission-critical tasks such as voice generation or transcription introduces third-party vulnerabilities and can cause potential service disruptions.

The question bank, while comprehensive for its current scope, requires regular updates to maintain relevance with evolving industry standards. The system's ability to handle highly specialized technical discussions remains constrained by the predetermined questions and expected responses.

Lastly, the current system does its best to emulate a human-led interview to ensure no additional biases are added to the interview results. However, the interviews still feel AI-led and not necessarily led by humans. There's a lot more scope to make this experience feel true to a real-world experience of giving the interview making it ideal for the candidates and companies in the long term.

References

1. E. Galanaki, A. Lazazzara, E. Parry, A cross-national analysis of e-HRM configurations: Integrating the information technology and HRM perspectives Organizing for digital innovation, Springer, Cham (2019), pp. 261-276
2. T. Bondarouk, C. Brewster Conceptualising the future of HRM and technology research International Journal of Human Resource Management, 27 (21) (2016), pp. 2652-2671
3. Piotr Horodyski, Applicants' perception of artificial intelligence in the recruitment process, Computers in Human Behavior Reports, Volume 11, 2023,100303, ISSN 2451-9588,https://doi.org/10.1016/j.chbr.2023.100303
4. B C Lee and B Y Kim, Development of an AI-Based Interview System for Remote Hiring, International Journal of Advanced Research in Engineering and Technology (IJARET), 12(3), 2021, pp. 654-663. http://www.iaeme.com/IJARET/issues.asp?JType=IJARET&VType=12&IType=3

5. Esch, P., Black, J. S., and Ferolie, J. Marketing AI recruitment: The next phase in job application and selection. Computers in Human Behavior, 90, 2019, pp. 215-222.
6. Suen, H. Y., Chen, M. Y. C., and Lu, S. H. Does the use of synchrony and artificial intelligence in video interviews affect interview ratings and applicant attitudes? Computers in Human Behavior, 98, 2019, pp. 93-101.
7. Nawaz, N. How far have we come with the study of artificial intelligence for the recruitment process? International Journal of Scientific & Technology Research, 8, 2019, pp. 488-493
8. Reddy, A. M., Yarlagadda, S., & Akkinen, H. (2021). An extensive analytical approach to human resources using random forest algorithm. arXiv preprint arXiv:2105.07855.
9. S. I. Horat, K. Can Kara and A. Karakaş, "Job Pre-Interview System with Artificial Intelligence," 2019 1st International Informatics and Software Engineering Conference (UBMYK), Ankara, Turkey, 2019, pp. 1-4, doi: 10.1109/UBMYK48245.2019.8965497.
10. C. Jayasekara et al., "Artificial Intelligence Agent to Identify the Correct Human Resources," 2023 5th International Conference on Advancements in Computing (ICAC), Colombo, Sri Lanka, 2023, pp. 424-429, doi: 10.1109/ICAC60630.2023.10417197.