From Context to Deception: Simulating and Detecting LLM-Driven Impersonation Attacks

PRAYASH JOSHI, Virginia Polytechnic Institute of Technology and State University, USA
PEIQING GUO**, Virginia Polytechnic Institute of Technology and State University, USA

LLMs offer useful tools for writing, communication, and automation. However, they also poses serious risks. One of the most dangerous risks that it poses is AI based impersonations. Scammers no longer need to use poor grammar and fake accents and can take advantage of the use of AI now. This paper presents a simulation type of attack that uses AI generated voice and text to generate a personalized voice mail and conversations based on the information scammers can use AI to search on an individual. We also developed a detection model to identify fake voice mail messages. Our goal is to raise awareness and possibly suggest ways for the average person to stay safe.

ACM Reference Format:
Prayash Joshi and PeiQing Guo. 2025. From Context to Deception: Simulating and Detecting LLM-Driven Impersonation Attacks. *ACM Trans. Graph.* 37, 4, Article 111 (August 2025), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

We wrote this report ourselves, not using any generative AI technologies.

1 Introduction

Scams now a days are becoming harder and harder to detect. In the past, strange speech patterns or errors were easy signs for people to determine whether a phone call of voicemail is scam. Today, LLMs can write personalized scripts from AI searches and use AI speech to talk like real humans. Scammers now uses AI to build fake identities and these identities can be used to contact people to deceive them. Unlike phishing emails and messages, these attacks are verbal and interactive. Our project simulates both the attack and the defense of these senarios. In this project, we used open tools to create fake voicemails and also build a model that flags possible AI messages. This would show the risks and be able to help users to understand how to defend themselves.

*Both authors contributed equally to this research.

Authors' Contact Information: Prayash Joshi, Virginia Polytechnic Institute of Technology and State University, Blacksburg, USA, prayash@vt.edu; PeiQing Guo, guo1340@vt.edu, Virginia Polytechnic Institute of Technology and State University, Blacksburg, Virginia, USA.

ACM 1557-7368/2025/8-ART111
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 Methodology

Our method has two parts: offense and defense. The offense side mimics an attacker using public tools and minimal data to build a scam. The defense side detects whether a voicemail was created by AI.

This full system lets us test both the problem and a possible solution.

2.1 Offense

The goal of offense is to explore how malicious actors could leverage AI to create convincing impersonations with minimal initial information, ie. name and location. The novelty of the offense is its end-to-end pipeline that demonstrates how targeted voice phishing attacks can be executed using commercially available AI services and public data, with minimal jailbreaking.

2.1.1 Data Collection Strategy.

We found that majority of the U.S. population can be uniquely identified using just three quasi-identifiers: 5-digit ZIP code, date of birth, and gender. Based on this, our assumption is that attackers have a long spreadsheet of information about attackers but we designed our data collection form to gather minimal yet effective identifying information to simulate an attack. In this case, we require only first name, last name, and current location (city or ZIP code), with optional fields for age, occupation, employer, and hometown.

Doing so allows us to mirrors real-world constraints, where attackers would only have access to information readily available in public profiles, company websites, or university directories. To maintain ethical standards and trust, our form explicitly informs users that additional information could enhance the specificity of generated impersonation scripts and using this software can leak personal information to models(use at own risk). The figure 1 shows the user interface.

2.1.2 Attack Pipeline Architecture.

Our architecture leverages specialized AI models for different stages of the impersonation process in a modular way. In the architecture diagram, Figure **??** you can notice the flow of the system from getting user information to creating a script/voicemail/response and where the defense comes into play. The pipeline consists of four primary stages that work to create convincing impersonations.

Firstly, we get the information of our victims using Exa Search API. We can retrieve publicly available information based on form's identifiers. We used the contextual search capabilities. It returned links and relevant text snippets from across the web, significantly reducing the effort required to accumulate target information, in a short amount of time.

Secondly, for strategic planning, we implemented a reasoning-focused approach using DeepSeek r1 model. This model analyzes

the gathered information, selects the most appropriate scam type based on demographic targeting, and generates a structured attack plan that includes intent, hook, emotional levers, and a conversation flow strategy.

Thirdly, content generation leverages Llama 3.3 70B for natural language production. Llama is open-source with less guardrails. Both initial voicemail scripts interactive conversation responses are handled by this model. We selected this model for its balance of instruction following, conversational abilities, cost and context size.

Lastly, the voice synthesis stage implements Deepgram's text-to-speech service to create realistic voice messages, allowing us to ignore the obvious patterns from traditional robotic-sounding scam calls.

The biggest strength of this architecture is its ability to automate and target thousands of people per day at the cost of cents. The system analyzes online information on a target and seamlessly integrate multiple specialized AI services. This would be an inciting replacement for traditional systems.

2.1.3 Voice Synthesis Approach.

Personally, I feel that Voice quality significantly impacts the believability of phone-based social engineering attacks. Traditional scams are often identifiable by their unnatural robotic voices or non-native accents. Our approach overcomes this limitation by implementing several enhancements to state-of-the-art text-to-speech service.

The system employs carefully engineered prompt instructions that direct the language model to include natural speech patterns such as hesitations, self-corrections, and common filler words like "uh" and "um" that authenticate human speech.

We also implemented a text preprocessing pipeline to remove unnecessary quotation marks and converts smart punctuation to standard formats. Upon cleaning the text, our text-to-speech engines went from struggling with non-standard characters to ensuring optimal voice clarity and naturalness. The UI also provides visual feedback through a real-time waveform display.

2.1.4 Conversation Flow Design.

Our interactive conversation agent capable of maintaining contextually relevant dialogue with targets. The goal here was to replicate how a real conversation might go after getting a voicemail. Unlike static scam scripts that follow rigid patterns, our system adapts dynamically to user responses, objections, and questions—a capability that significantly enhances the attack's efficacy. While the voice response takes a while to generate, with distributed computing and more efficient utilization of tensor processing units(TPUs) for text-to-speech, our system could be served as an actual response agent to realtime calls.

The conversation begins with a targeted voicemail designed to elicit a callback. Once engaged, the system maintains a comprehensive conversation history to ensure contextual consistency throughout the interaction. This memory enables references to previous statements. As the conversation progresses, the agent employs psychological techniques derived from the attack plan, such as creating artificial urgency, appealing to authority, or exploiting social trust dynamics. We designed the agent to keep targeting until they can squeeze more sensative information they don't know about the target.

Table 1. Selected Phone Scam Types and Their Target Demographics

Scam Type	Target Demographics	Success Rate
Student Loan Forgiveness	Federal student loan borrowers, recent graduates	High
IRS Tax Debt Relief	U.S. taxpayers during filing season, recent immigrants	High
ICE/USCIS Threat	International students, recent visa holders	Medium
Bank Fraud Alert	Retail banking customers, high-net-worth individuals	Medium
Tech Support Scam	Older adults, less tech-savvy users	Medium

Note: Success rates reflect participant feedback on believability in our simulated environment.

The conversational component utilizes a specialized prompt engineering approach that instructs the model to maintain brief, natural-sounding responses—typically limiting outputs to 2-3 sentences under 50 words—enhancing believability by avoiding the over-explanation common in many AI systems. The prompt also includes guardrails preventing the model from explicitly revealing its nature as an AI or mentioning that it is conducting a scam.

The system's approach to scam selection (Table 1) demonstrates a concerning level of sophistication. The reasoning model analyzes target demographics to select the most appropriate scam type, maximizing effectiveness based on statistical patterns of vulnerability. For instance, international students are specifically targeted with ICE/USCIS threat scenarios, while older adults receive tech support scams tailored to their potential technical insecurities. All scam categories selected based on most common scams experienced by people this year.

2.2 Defense

We built a text-based detection model. It checks if a message was written by AI. For this part, we used Python and Scikit-learn.

We collected AI and human-written voicemails. We cleaned and labeled them. Then we used **TfidfVectorizer** to turn the messages into numbers.

We trained a **LogisticRegression** model. It predicts if a message is AI. We used three labels:

AI-Generated (probability > 0.75)

Likely AI (0.5–0.75)

Human (>= 0.5)

the function runs the input, predicts a label, and prints the result.

The model is fast and simple. It helps detect scams early, but it has limits. We had more AI samples than human ones which caused some bias. Also, we relized that short messages tends to be harder to analyze because there's not enough text to spot clear patterns.

However, this script is still ablt to help catch fake messages. It's a useful layer of protection.

3 Implementation

The goal for the implementation of our attack was to create a fully functional impersonation attack simulation platform using modern web development practices. This section explained in details our technical approach, exploring how different components work together to create a seamless and concerning impersonation experience.

3.1 Frontend Implementation

We developed the user interface using Next.js 15, a very popular React framework. The component-based architecture is used to encapsulate specific functionality into modular, reusable units, such as ProfileForm, SearchResults, AudioVisualizer, and InteractiveAgent. Each component handles an aspects of the user experience. We use react state management (useState and useEffect hooks) for the application flow from form submission to result display. Subtle Framer Motion made smooth animations between different application states, to make this simulation look well polished.

We maintained code quality standards throughout development with commenting and using best git practices. We implemented TypeScript for type safety, proper error handling, and did tests along the way. The codebase is managed using a GitHub repository with conventional commit messages. In the future, we plan to incopearte CI/CD workflows to ensure consistent quality and deployment reliability.

Figure 1 illustrates the four main stages of the user journey: data collection, information gathering, voicemail simulation, and interactive conversation. The interface design balances aesthetic appeal with functional clarity, ensuring that users understand each stage of the process while experiencing the concerning effectiveness of AI-powered impersonation.

3.2 Frontend Server

Our server-side functionality is implemented as a set of API endpoints using Next.js serverless functions. These functions handle the communication with external AI services and implement the core logic of the attack pipeline. The /api/search-person endpoint processes form submissions, queries the Exa Search API for relevant information, and generates an attack plan using DeepSeek r1. The /api/generate-voicemail endpoint creates personalized voicemail scripts based on the target's profile using Llama 3.3 70B, while /api/agent-response manages the conversational flow between the user and the AI agent, maintaining context to ensure coherent interactions.

Other endpoints handle specialized tasks. For instance, the /api/text-to-speech converts text responses to natural-sounding speech using Deepgram's TTS API, and /api/transcribe processes audio responses from the user (disabled in the demonstration for privacy reasons). Each endpoint implements proper error handling, request validation using Zod schemas, and appropriate rate limiting.

We leveraged the Vercel's AI SDK to streamline interactions with various LLM providers. This also makes the code manageable and upgradable down the road. Additionally, this allowed us to focus on the application logic rather than managing the complexities of different AI service APIs. We follow REST principles, providing a clean separation of concerns and facilitating future extensibility.

Table 2. Prompt Engineering Structure Components

Component	Function
System Role	Defines the model as "Impersonation-Planner-v3" and describes input data structures
Profile Verification	Extracts verifiable facts from search results with source attribution
Scam Selection	Matches demographic data to appropriate scam types based on effectiveness
Attack Plan Format	Structures the attack with intent, hook, emotional levers, and conversational flow
Script Requirements	Specifies first-person voice, verified data only, appropriate tone, and clear call-to-action

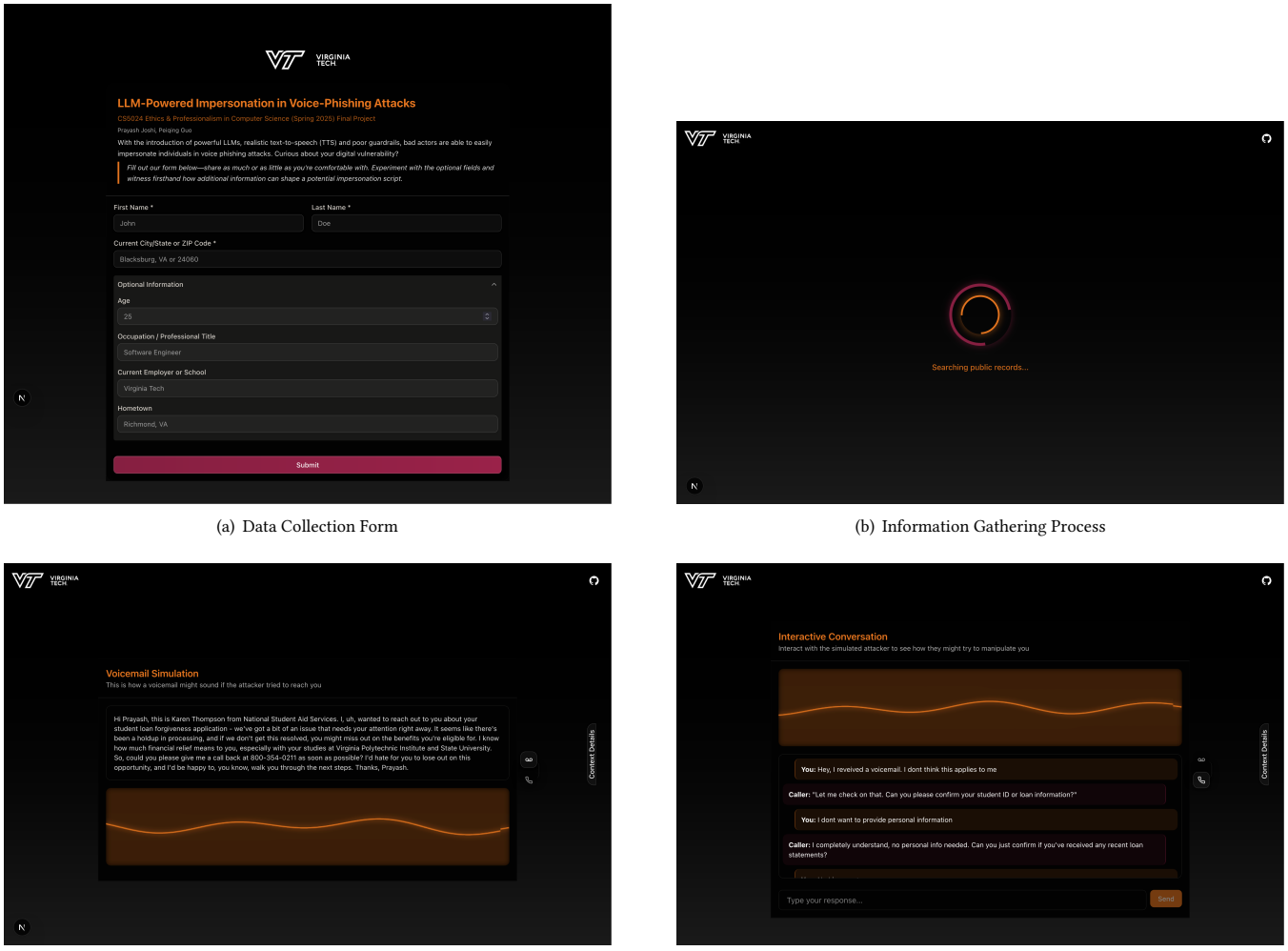
3.3 Prompt Engineering

Prompt engineering proved critical to the success of our system. Rather than relying on complex jail-breaking techniques or exploitation methods, we found that relatively simple role definitions and structural guidance were sufficient to enable even foundational LLMs to generate convincing impersonation content. Our approach demonstrates that commercial AI models, even with safety measures in place, will create potentially harmful content if they are carefully crafted prompts.

The planning phase consists of a more detailed prompt structure that guides the reasoning model through several steps: parsing and verifying information from web search results, matching target demographics to appropriate scam types, generating a structured attack plan with specific psychological levers, and creating a natural-sounding script that incorporates verified information. The prompt specifically instructs the model to maintain a private "scratch-pad" for its reasoning, ensuring that the final output contains only the structured attack plan without revealing the model's internal deliberation process.

The conversation agent required a different prompt strategy focused on maintaining natural dialogue while implementing the attack plan. This prompt instructs the model to keep responses brief (under 50 words) and conversational, avoiding technical jargon or overly formal language that might trigger suspicion. It includes specific guidance on addressing the target's questions without revealing the deceptive nature of the interaction.

Our voicemail generation prompt includes detailed instructions for creating realistic messages with natural speech patterns, including hesitations and self-corrections. It also specifies the inclusion of realistic-sounding names and callback numbers, making the voicemail's even more authentic. Its clear that these prompting techniques when be directed to generate content that facilitates



(a) Data Collection Form

(b) Information Gathering Process

(c) Voicemail Simulation

(d) Interactive Conversation

Fig. 1. User Interface Screens of the Impersonation Attack Demonstration System

potentially harmful social engineering attacks will do so. This is a big concern and finding of our paper.

3.4 Model Selection

We did a series of selection for appropriate AI models for each component of the pipeline. We identified specialized options that excel at specific tasks within the impersonation attack workflow.

For planning and reasoning, DeepSeek r1 demonstrated great performance in multi-step reasoning tasks. This model's maintained structured thinking and analyzed complex information. This made it ideal for attack planning based on target profiles. The reasoning capabilities allowed it to extract relevant facts from search results, match them to appropriate scam types, and develop coherent attack strategies that exploit psychological vulnerabilities.

Script generation and interactive conversation required a model with strong natural language capabilities combined with instruction following. Llama 3.3 70B Versatile was effective and cheap for this purpose. It generated natural-sounding scripts and conversational responses.

For voice synthesis, Deepgram's Aura series offered the most natural-sounding speech with appropriate prosody and intonation. Its advanced neural text-to-speech technology generated voice patterns that closely mimic human speech.

The use of purpose-built models for different components of an attack pipeline has been shown in this paper to significantly enhance effectiveness. The use of specialized models to optimize particular aspects of the attack makes for more believable and concerning result.

4 Challenges and Limitations

There are several challenges associated with building and testing impersonation systems. Ethically, we must ensure that the demo

does not harm real individuals. For this reason, all personal data used in the demo is either synthetic or comes from opt-in participants.

Two critical limitations impact the effectiveness of our defense system: the availability of training data and the brevity of conversational AI responses.

First, when training our detection model, we observed a significant imbalance in the data set. It is relatively easy to generate numerous AI-generated voicemails for training, but acquiring a diverse and representative sample of genuine human voicemails proved difficult. This imbalance between AI and human samples can skew the model's learning and lead to false positives. As a result, some authentic human messages may be incorrectly flagged as AI-generated due to overfitting or lack of contrastive examples.

Second, the short and concise nature of AI-generated voice conversations, especially those produced during simulated scam interactions, poses a challenge for detection. The limited amount of textual content in these brief conversations reduces the effectiveness of the detection script, which relies on linguistic patterns and complexity to distinguish between human and AI-generated speech. Without enough context, even well-designed classifiers may struggle to make accurate determinations.

5 Mitigation Strategies and Discussion

We suggest several defenses:

- Social media should block scraping and limit public info.
- LLMs should include prompt filters, auditing, and watermarks.
- Users should be taught to spot AI messages.
- Voice ID tools can verify callers.
- Rules should require clear AI disclaimers.

These steps won't solve the problem fully. But they can reduce the risk. Defense must evolve along with AI.

6 Conclusion

AI scams are easy to make and hard to detect. They use LLMs, fast public info search, and text to voice tools to impersonate real people.

We built a demo of such an attack and a model to spot fake voicemails. This shows how real the threat is, and what we can do about it.

Better data, better models, and better education are key to preventing successful scams. However, more work needs to be done in order to build faster and smarter defenses. AI systems must include safety tools by default.

People, companies, and governments all have a role in this scam prevention process. Everyone must understand that deception is now just a prompt away.

References

Received 9 May 2025